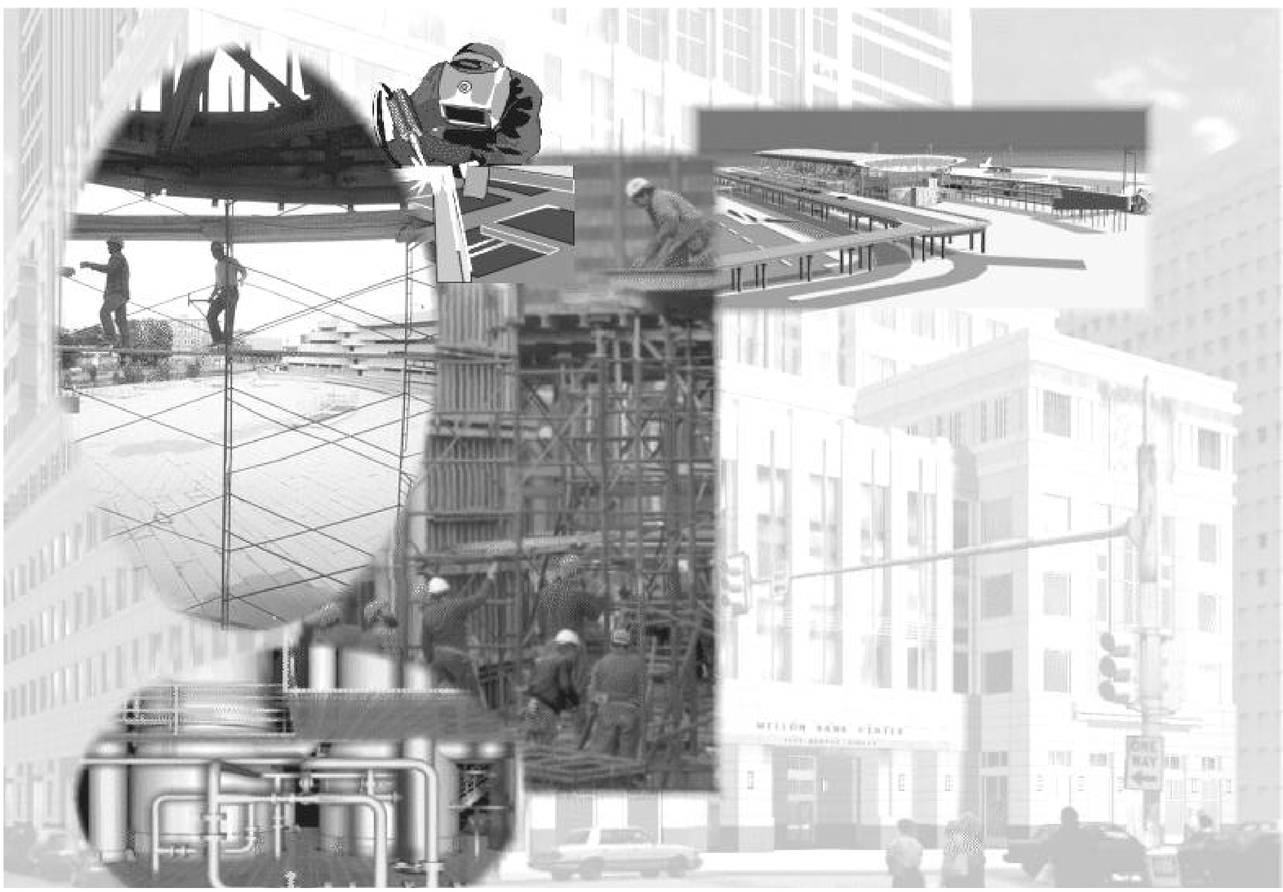
Industry Foundation Classes - Release 2.0

Projects Documentation

- Project Summaries
- AEC Process Definitions
- IFC Model Requirements Analysis



Draft 2

February 24, 1996

International Alliance for Interoperability

Enabling Interoperability in the AEC Industry

Table of Contents

1. Release 2.0 Projects Overview

1.1 Projects Overview Spreadsheet

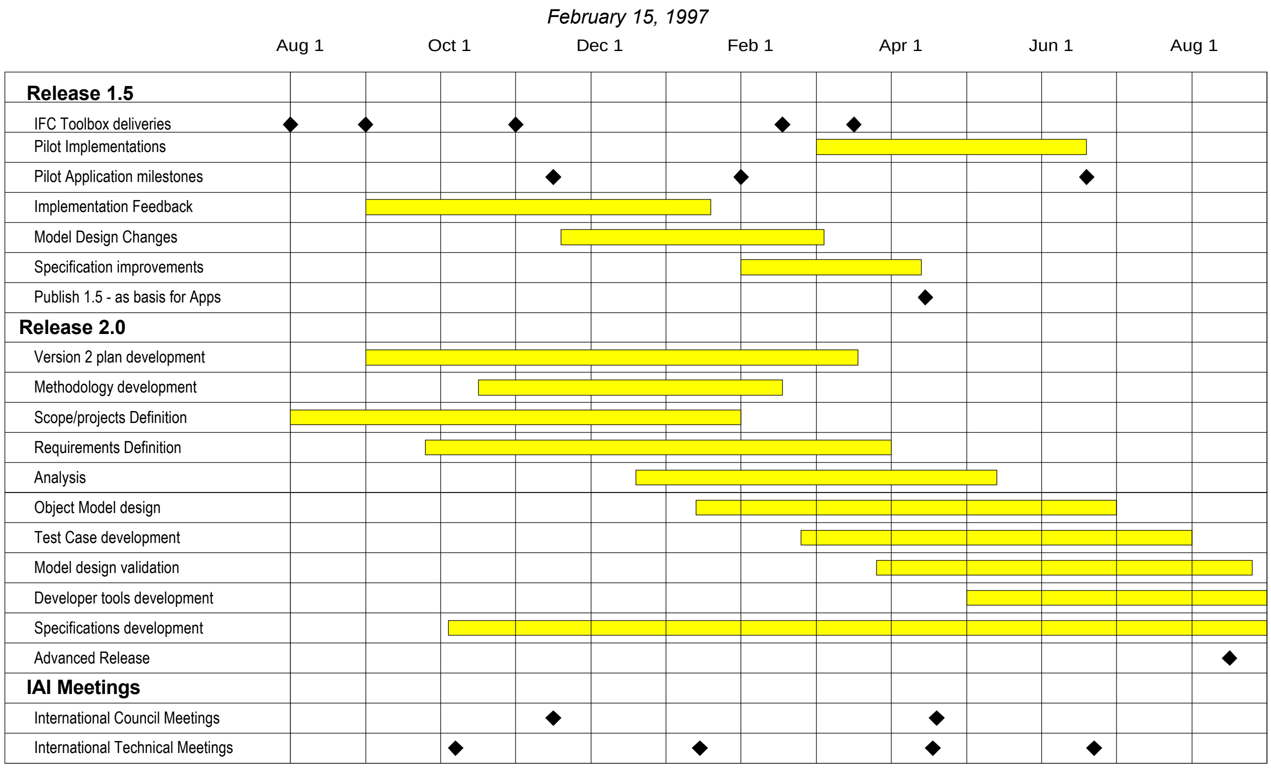
Note: in the "ProJ Grp" column below → "A" means that the project will be included in Release 2.0 as originally proposed, "B" means that the scope has been reduced from what was originally proposed, "C" means that the project has been delayed for inclusion in Release 3.0.

							-----Docs ---		
Prio	Proj Grp	Proj ID	AEC Domain Grouping	Project	Size	Size	PS	US	AN
			Project description	Lead	(0-5)	(0-5)	(0-5)	(0-5)	(0-5)
			Sub-Project description		Domain Group Work	Tech. Support Work	State of Doc	State of Doc	State of Doc
			Architecture						
7	A	AR-1	**Completion of the Architectural Model**	NA - Ken Herold iaiexec@interoperability.com	5	5	4	4	4
			Building Core design						
			Building Shell design						
			Roof Design						
			Stair Design						
			Publilc Restroom design						
			Conceptual Space Planning						
1	B	AR-2	**Space planning for escape routes**	UK - Jay Patankar patankar@dial.pipex.com	2	1	3	2	3
			Building Services						
7	B	BS-1	**HVAC Systems design**	NA - Jim Forester forester@crl.com	4	2	4	4	4
			HVAC duct design(air-side delivery systems)						
			HVAC hydronic design(water-side delivery systems)						
4	B	BS-2	**Power and Lighting systems design**	UK - Tony Baxter hevacomp@hevacomp.demo n.co.uk	4	2	0	0	0
			Lighting design						
			Power design						
6	B	BS-3	**Pathway coordination with other trades**	D - Robert Rotternann 100041.2347@compuserve.c om	2	2	0	0	0
6	B	BS-4	**Thermal loads calculation**	D - Mr. Tonke 100436.705@compuserve.co m	1	1	0	0	0
			Client Briefing (Building Owners)						
3	C	CB-1	**Process Model (information supplied to/by clients)**	UK - Matthew Bacon baafplan@cix.compulink.co.u k	4	1	0	0	0

			Construction & Construction Management							
2	C	CM-1	**Procurement and Logistics**		UK - Graham Storer g-storer@tel-consult.co.uk	3	2	0	0	0
1	C	CM-2	**Temporary Construction Elements**		J - Takayori Takamoto takamoto@kke.co.jp	3	2	0	0	3
			Codes and Standards							
2	B	CS-1	**Code compliance checking**		NA - Robert Briggs rs_briggs@pnl.gov	3	2	4	4	4
				Code Compliance - Energy						
	B	CS-2	**Code extensions**		S - Mr. Wong Wai Ching	4	2	3	0	3
				Code Compliance - Disabled Access						
				Code Compliance - Escape routes						
			Estimating and Scheduling							
5	A	ES-1	**Cost Estimating**		NA - Mike Cole mikec@timberline.com	3	1	4	4	4
				Cost Item Identification						
				Task & Resource Modeling						
				Cost Modeling						
			Facilities Management							
3	C	FM-1	**Engineering Maintenance** (periodic equipment maint)		UK - Mike Goodman +44 117 943 4113	2	1	0	0	0
1	C	FM-2	**Architectural Maintenance** (painting, roof sys., walls, etc.)		N - Arto Kiviniemi arto.kiviniemi@vtt.fi	2	1	3	0	0
2	B	FM-3	**Property Management** (building owner viewpoint)		N - Poul Sorgenfri Ottosen	4	2	3	3	3
4	A	FM-4	**Occupancy Planning** (moving people around)		NA - Kevin Yu yu@civil.ubc.ca	3	2	4	4	4
				Move Planning						
				Workstation design						
				Wrokstation layout in an open plan						
			Simulation							
3	A	SI-1	**Photorealistic Visualization**		NA - Vladimir Bazjanac vlado@gundog.lbl.gov	2	1	4	4	4
			Structural Engineering							
6	B	ST-1	**Steel Frame construction**		UK - Phil Jackson pmj@mm-croy_mottmac.com	5	4	0	0	0
				General structural model						
				Structural Steel Frames						
1	C	ST-2	**Concrete Frame construction**		F - Patrice Poyet poyet@cstb.fr	3	2	0	0	0
1	C	ST-3	**Substructure design(foundations, etc.)**		J - Takayori Takamoto takamoto@kke.co.jp	3	3	0	0	3
1	C	ST-4	**Load Definitions/Rating (info for FM)**		D - Dr. Dietrich rdietrich@hlzm.de	1	1	0	0	0
			X-Domain (CORE) model features							
6	C	XM-1	**Referencing external Libraries (product data,etc.)**		UK - Patrick Barbour 100342.2537@compuserve.c	4	3	0	0	0

						om					
5	B	XM-2		**Project document management**		NA - Ray Brungard rbrungard@tcco.com	5	4	4	4	3
7	A	XM-3		**Overall model features/architecture extension**		STF - Thomas Liebich thomas@cab1.m.eunet.de	5	5	0	0	0
					General Network model						
					General purpose constraints (design, codes, alignment)						
					Semantic associations (element aggregator)						
					General purpose tables						
			Projects distribution								
				By AEC discipline/domain							
				2	Architecture						
				4	Building Services						
				1	Client Briefing						
				2	Codes and Standard						
				2	Construction						
				1	Estimating/Scheduling						
				4	Facilities Management						
				1	Simulation						
				4	Structural						
				3	X-domain						
				24	Total projects						
				By chapter lead							
				1	French (F)						
				3	German (D)						
				2	Japan (J)						
				2	Nordic (N)						
				7	North America (NA)						
				1	Singapore (S)						
				7	United Kingdom (UK)						
				1	Spec Task Force (STF)						
				24	Total projects						

1.2 Project Schedule



2. Release 2.0 Project Summaries

Architecture

2.1 AR-1 Completion of the Architectural Model

2.1.1 Project Description

The Architectural Domain with tackle six processes (listed below) that are grouped under completion of the Architectural Model. To find out more about the proposed processes, review the Architectural project proposal document. The processes to be included in the 2.0 release will cover processes that span from the Schematic design phase of Architecture through refinement in the Construction Document phase. :

1. Core Design
1A. Stair Design
1B. Restroom Design
2. Roof Design
3. Shell Design
4. Block & Stack

2.1.2 Project Team

Project Leader à Ken Herold - North American

Chapter	Name	Company	Email	Hrs / Week
NA	Gustavo A. Lima	Cannon	glima@cunnon.com	
	Bill O'Malley	Hammel Green and Abrah...	BOMalley@EMAIL.HGA.COM	
	Barbara Golte...	Heller & Metzger, PC	74212.354@compuserve.com	
	Ken Herold	HOK	iaiexec@interoperability.com	180 H
	Steve Stevens	Intergraph	festeven@ingr.com	104 Hr
		Intergraph		
	Juniper Russell	Juniper Russell & Assoc.	juniper@novimundi.com	
	Ed Ebbing	MC2	eebbing@mc2-ice.com	68 Hr
	Martin Rozmanith	RMW Architecture + Design	marty_rozmanith@rmw.com	
	Ardie Aliandust	RTKL	2350@la.rtkl.com	104 Hr
	Bill Houstoni	RTKL	bhouston@balt.rtkl.com	
	Nick Reveliotty	The Kling Lindquest Pa...		
	Tony Sinisi	The Kling Lindquest Pa...	76636.1043@compuserve.com	
	Beth Brucker	USA-CERL	B-Brucker@cecer.army.mil	104 Hr
	Paul Lewis	Visio	paull@visio.com	
	Rob Wakeling	Visio	robw@risiu.com	
German				
UK				

Nordic				
Singapore				

2.1.3 Scope of Work

AEC processes to be supported	-	6	Est. total AEC expert time (days)	-	30.2
Expected IFC Model Impact (1 *(min)* to 5)	-	4	Est. total Info Modeling expert time (days)	-	nn
Degree of technical difficulty (1 *(min)* to 5)	-	4	Est. total Software/PM expert time (days)	-	nn

2.1.4 Resources Required / Committed

Member Company Resources	**Reqired Days**	**Market Value**		**Days Committed**	**Resource shortfall**
Process Model	30	$10,195		nn	nn
Usage Requirements	30	$10,195		nn	nn
Object Model development	30	$10,195		nn	nn
Integration	7.5	$2,600		nn	nn
Test Case development	37.75	$12,740		nn	nn
Implementation technical support	7.5	$2,600		nn	nn
Management and Review	7.5	$2,600		nn	nn
Total Member Company Resources	**151**	**$51,000**		nn	nn
Travel		**$68,000**			
Project Support	**Required Days**	**Market Value**			
Technical support	nn	$nn			
Project management	nn	$nn			
Publication and Administration	nn	$nn			
Equipment and software	nn	$nn			
Travel and subsistence	nn	$nn			
Total Project Support	**nn**	**$nn**			
Total for Project	**nn**	**$nn**			

2.2 AR-2 Space Planning for Escape Routes

2.2.1 Project Description

This project includes the following four processes:

- Means of Escape from Spaces

The person responsible for planning means of escape takes the brief data and establishes various types of occupancy within the project envelope. This is then divided into usage compartments from the point of view of the client. The clients usage compartments are then subjected to regulation and compartments derived which conform to regulatory constraints. By default safe zones are the remaining spaces. Escape routes, either to another safe compartment or to outside air are then defined by linking safe spaces.

2.2.2 Project Team

Project Leader à Jay Patankar UK

Chapter	Name	Email	Hrs / Week
UK	Jay Patankar	patankar@dial.pipex.com	
UK	Steve Race	darcyrace@dial.pipex.com	
Total for Project			
Total person-days			

2.2.3 Scope of Work

AEC processes to be supported	-	3	Est. total AEC expert time (days)	-	30
Expected IFC Model Impact (1 *(min)* to 5)	-	5	Est. total Info Modeling expert time (days)	-	40
Degree of technical difficulty (1 *(min)* to 5)	-	4	Est. total Software/PM expert time (days)	-	nn

2.2.4 Resources Required / Committed

Member Company Resources	**Reqired Days**	**Market Value (apply $45/hr-head)**		**Days Committed**	**Resource shortfall**
Requirements definition					
Process Model	30	£13200			
Usage Scenaria	20	£8800			
Model design					
Object Model development *(w/ tech.Support)*	40	£17600			
Integration *(w/ tech.Support)*	10	£4400			
Design and Implementation validation					
Test Case development	20	£8800			
Review/feedback on implementations	40	£17600			
Project Management					
Project management and administration	34	£14960			
Travel and Meetings	80	£35200			
Total Member Company Resources	**274**	**120560**			

Model/Specification development support	**Required Days**	**Market Value**			
Technical support	50	£12500			
Project management	24	£10500			
Publication and Administration	10	£2200			
Equipment and software		£2000			

Travel and subsistence		£2000			
Total Project Support		**£29200**			
Total for Project		**£149760**			

Building Services

2.3 BS-1 HVAC System Design

2.3.1 Project Description

This Project includes the following processes:

- HVAC Duct Design
- HVAC Hydronics Design

These processes will involve utilizing the network classes defined in the IFC 2.0 Core model. This effort will be led by the North American Building Services Committee, but will be an international collaborative effort. This will ensure that the resulting system design extensions are globally applicable.

Engineers responsible for the design of duct and hydronic systems may be consulted during the building conceptual stage. However, the major design effort occurs after the architect has substantially completed the building drawings. The design process includes both the schematic and detailed description of duct and hydronic components. These components include sections of duct and pipe, fittings, accessories such as dampers, valves, and terminals. This process also includes the connection of these components to equipment such as fans and pumps. Classes for equipment were defined in IFC Version 1.x, and are not elaborated in this proposal. The system design process also includes construction cost estimates but actual costs are determined by contractors using drawings and specifications prepared by the Building Services engineers.

Significant cost savings will result from the application of IFC's to systems design in Building Services.

- Building geometry and construction materials used in the design of HVAC load calculations and the fluid distribution systems.
- The exchange of data between engineering design and analysis programs with manufacturers' equipment selection programs.
- The production of schedules of bill of materials for the system components.
- Producing the data for engineers cost estimates and for contractors actual construction cost estimates.
- The opportunity for integration of control components used for the operation of these systems.

2.3.2 Project Team

Project Leader à Jim Ahart (Domain) Jim Forester (Technical)

Chapter	*Name*	*Email*	*Hrs / Wk*
NA	Jim Ahart	103073.1120@compuserve.com	8
NA	Tim Baliles	tbaliles@cellabarr.com	6
UK	Tony Baxter	100316.3252@compuserve.com	4
NA	John Deal	75601.1346@compuserve.com	4
NA	Rod Dougherty	rod.dougherty@landis+gyr.sprint.com	6
NA	Tom Edman	tom@htc.honeywell.com	4
NA	Jim Forester	jim@marinsoft.com	6

NA	Scott Frank	sfrank@pipeline.com	4
NA	Jim Lindquist	jlindquist@tklp.com	4
NO	Pekka Metsi	pekka.metsi@granlund.fi	4
FR	Jean-Luc Monceyron	monceyron@cstb.fr	4
NA	John Murphy	jmurphy@trane.com	4
FR	Patrice Poyet	poyet@cstb.fr	4
D	Robert Rotterman	100041.2347@compuserve.com	4
NA	Larry Schaefer	larry.schaefer@carrier.wltk.com	4
NA	Tony Sherfinski	tony.sherfinski@greenheck.com	4
D	Jeremy Tammik	73174.2355@compuserve.com	4
UK	Jeff Wix	100342.2537@compuserve.com	4
Total for project			***82***

2.3.3 Scope of Work

AEC processes to be supported	2	Est. total AEC expert time	8 wks
Expected IFC Model Impact (1 to 5)	3	Est.total Info Modeling time	8 wks
Degree of technical difficulty(1 to 5)	3	Est. total Software/PM expert time	8 wks

2.3.4 Resources Required / Committed

Member Company Resources	**Reqired Days**	**Market Value**		**Days Committed**	**Resource shortfall**
Requirements definition					
Process Model	10	$12000		10	0
Usage Scenaria	15	$18000		10	5
Model design					
Object Model development *(w/ tech.Support)*	10	$12000		15	15
Integration *(w/ tech.Support)*	20	$24000		?	?
Design and Implementation validation					
Test Case development	15	$18000		10	20
Review/feedback on implementations	15	$18000		?	?
Project Management					
Project management and administration	15	$18000		30	0
Travel and Meetings	10	$12000		10	0
Total Member Company Resources	**110**	**$132000**		**85+**	40+
Model/Specification development support	**Required Days**	**Market Value**			
Technical support	5	$6000			
Project management	10	$12000			
Publication and Administration	10	$12000			
Equipment and software	5	$6000			
Travel and subsistence	10	$12000			
Total Project Support	**40**	**$48000**			

Total for Project	**150**	**$180000**				

2.4 BS-2 Power and Lighting Systems Design

{{ Project Summary for this project not yet available }}

2.5 BS-3 Pathway Design and Coordination

{{ Project Summary for this project not yet available }}

2.6 BS-4 HVAC Loads Calculation

{{ Project Summary for this project not yet available }}

Client Briefing

2.7 CB-1 Client Briefing

{{ This project has been delayed for inclusion in Release 3.0 }}

Construction / Construction Management

2.8 CM-1 Procurement and Logistics

{{ This project has been delayed for inclusion in Release 3.0 }}

2.9 CM-2 Temporary Construction

{{ This project has been delayed for inclusion in Release 3.0 }}

Codes and Standards

2.10 CS-1 Code Compliance Enabling Mechanism/Energy Code Compliance Checking

2.10.1 Project Description

This project has two parts: CS-1A - ***Code Compliance Enabling Mechanism*** and CS-1B ***Energy Code Compliance Checking***. These two parts have been combined into a single project for administrative efficiency. Part A of the project will define a generic code compliance enabling mechanism that will be applicable to codes of various types; e.g., accessibility, egress, and energy. The mechanism will likely involve defining new abstract classes for code compliance in the core model. Part A will be an international collaborative effort, which will ensure that the resulting enabling mechanism is broadly applicable. Part B, Energy Code Compliance, will serve an important role in validation of the generic mechanism for a set of code applications. This work will be performed primarily by the North American Chapter and will enable established energy code compliance applications to be made IFC compliant.

Code compliance checking is performed by building designers, systems designers, and code enforcement officials. Compliance with codes begins during the programming phase when designers determine which codes apply to the building project. Preliminary code reviews are frequently performed during schematic design, and more thorough reviews are performed by members of the design team late in the design process before construction documents are complete. Building code officials perform plan reviews as part of the building permit process. Designers and code official perform drawing dimension takeoffs as necessary to ensure compliance. Information about building systems, assemblies, layout, etc. is gathered during this process and compared to the requirements for each applicable code.

Codes impact virtually all disciplines involved in building design and construction processes, and code considerations persist throughout a building's life cycle. Energy codes are strongly related to architectural, HVAC, and electrical design processes. While it would be difficult to establish a reliable estimate of time and cost savings from IFC support of code checking, the tedious nature of code review and the large cost and schedule impacts that code violations can cause suggest that there will be high demand for code checking applications. Energy codes represent an attractive application for IFC support because of their extensive requirements for building data that are already in electronic form (e.g., geometric data and lighting fixture data) and demonstrated strong demand--thousands of copies of these applications currently in use.

2.10.2 Project Team

Project Leader à Rob Briggs - North America Chapter

Chapter	*Name*	*Email*	*Hrs / Wk*
North America	Rob Briggs	rs_briggs@pnl.gov	10
Singapore	Tan You Tong	youtong@iti.gov.sg	2
France	Philippe Debras	debras@cstb.fr	2
UK	Robert Amor	trebor@bre.co.uk	1
Total for project			***15***

2.10.3 Scope of Work

AEC processes to be supported	-	1	Est. total AEC expert time (days)	-	5
Expected IFC Model Impact (1 *(min)* to 5)	-	2	Est. total Info Modeling expert time (days)	-	2

Degree of technical difficulty (1 *(min)* to 5) - 3 Est. total Software/PM expert time (days) - 2

2.10.4 Resources Required / Committed

Member Company Resources	**Reqired Days**	**Market Value**		**Days Committed**	**Resource shortfall**
Requirements definition					
Process Model	5	$4,000		5	0
Usage Scenaria	5	$4,000		5	0
Model design					
Object Model development *(w/ tech. Support)*	10	$8,000		10	0
Integration *(w/ tech. Support)*	8	$6,400		8	0
Design and Implementation validation					
Test Case development	5	$4,000		5	0
Review/feedback on implementations	5	$3,840		5	0
Project Management					
Project management and administration	5	$4,000		5	0
Travel and Meetings	5	$7,000		5	0
Total Member Company Resources	**48**	**$41,240**		48	0

Model/Specification development support	**Required Days**	**Market Value**		**Days Committed**	**Resource shortfall**
Technical support	nn	$nn			
Project management	nn	$nn			
Publication and Administration	nn	$nn			
Equipment and software	nn	$nn			
Travel and subsistence	nn	$nn			
Total Project Support	**nn**	**$nn**			
Total for Project	**nn**	**$nn**			

2.11 CS-2 Code Compliance Extensions

- Code Compliance - Disabled Access
- Code Compliance - Escape Routes

2.11.1 Project Description

The project covers specific application of the code compliance enabling mechanism (R2_CS-1) in serving the disabled access and escape routes code compliance.

Disable access code compliance is a process of assessing whether **the access provisions and facilities** of a building complies with one or more codes or standards **that serve the needs of the wheelchair user and ambulant disabled** enforced by various codes and standards promulgation entities.

Escape route code compliance is a process of assessing whether **the exit provisions and facilities** of a building complies with one or more codes or standards **that provide safe means of escape for occupants** enforced by various codes and standards promulgation entities.

The processes are performed by building designers and code enforcement officials during early design and submission stages, respectively. Automatic code compliance software based on the IFC models created in this project will help building designers to carry out self-checking of their designs in order to detect code violations as early as possible while design changes are still relatively cheap to make. Similarly, it also help the code enforcement officials to verify the plans submitted by the designers for building approvals.

The resources required to produce the IFC model for the disabled access and escape route are estimated to be 160 man-days over 20 elapse calendar weeks. Based on market value of $200 (Singapore) per man-days, a total of $32000 is required for the project.

2.11.2 Project Team

Project Leader à Mr. Wong Wai Ching - Singapore

Disable Access

Chapter	Name	Email	Hrs / Week
Singapore	Mr. Wong Wai Ching (leader)	- (through keewee@ncb.gov.sg)	2
Singapore	to be appointed (domain)	-	4
Singapore	Mr. Zhong Qi (info modeling)	zhongqi@iti.gov.sg	22
Singapore	Mr. Liew Pak San (software)	paksan@ncs.com.sg	4
F	??	??	??

A total of 32 man-hrs/week is required which is equivalent to 4 man-days/week (based on 8 hrs/days). Over 20 calendar weeks, a total of 80 man-days is required.

Escape Route

Chapter	Name	Email	Hrs / Week
Singapore	Mr. Wong Wai Ching (leader)	- (through keewee@ncb.gov.sg)	2
Singapore	to be appointed (domain)	-	4
Singapore	Ms Gosselin Yveline (info modeling)	gosselin@iti.gov.sg	22
Singapore	Mr. Liew Pak San (software)	paksan@ncs.com.sg	4
UK	??	??	??

A total of 32 man-hrs/week is required which is equivalent to 4 man-days/week (based on 8 hrs/days). Over 20 calendar weeks, a total of 80 man-days is required.

2.11.3 Scope of Work

AEC processes to be supported	- 2	Est. total AEC expert time (days)	- 15
Expected IFC Model Impact (1 *(min)* to 5)	- 2	Est. total Info Modeling expert time (days)	- 55
Degree of technical difficulty (1 *(min)* to 5)	- 2	Est. total Software/PM expert time (days)	- 10

2.11.4 Resources Required / Committed

Member Company Resources	**Required Days**	**Market Value**		**Days Committed**	**Resource shortfall**
Process Model	4	S$200		4	0
Usage Requirements	20	S$200		20	0
Object Model Development	80	S$200		80	0
Integration	20	S$200		20	0
Test Case Development	10	S$200		10	0
Implementation Technical Support	10	S$200		10	0
Management and Review	16	S$200		16	0
Total Member Company Resources	**160**	**S$32000**		**160**	**0**

Model/Specification development support	**Required Days**	**Market Value**		**Days Committed**	**Support shortfall**
Technical support	??	$??			
Project management	??	$??			
Publication and Administration	??	$??			
Equipment and software	??	$??			
Travel and subsistence	??	$??			
Total Project Support	**nn**	**$??**			
Total for Project	**nn**	**$nn**			

Estimating and Scheduling

2.12 ES-1 Cost Estimating

2.12.1 Project Description

This project is designed to increase the ability of the model to support cost estimating. The model already supports cost estimating to some degree. This project focuses refining and expanding that capability.

Most of the information used by cost estimating will be entered into the model by earlier design processes. At various times during the evolution of the design, an estimator will use the model to do cost estimating. During early design stages, very little information will be available, and only a rough estimate will be possible. As the model becomes more detailed, more accurate estimates are possible. When different designs are under consideration, “what if” or “alternate” estimates may be used to compare their cost impact. After a design and estimate are approved, inevitably, changes will be proposed and “change order” estimates will be required to determine the cost impact of the proposed change.

Using the IFC Model to do cost estimating saves time by using information provided by the design processes. It can also save time by making the task and resource data that it creates available to later processes such as scheduling. Using the model as the primary information source for estimating can also reduce errors and omissions that occur when data is entered into an estimating system by hand.

2.12.2 Project Team

Project Leader à Mike Cole - North American Chapter

Chapter	Name	Email	Hrs / Wk
NA	Mike Cole	mikec@timberline.com	10
NA	Ray Brungard	rbrungard@tcco.com	.5
UK	Jeffrey Wix	10342.2537@compuserve.com	?
NA	Peggy Woodall	peggy@bsdsoftlink.com	?
NA	Annette Stumph	a-stumpf@cecer.army.mil	?
NA	Roger Grant	rgrant@rsmeans.com	?
DE	Hans-Peter Sanio	San@mail.rib.de	?
Total for project			

2.12.3 Scope of Work

AEC processes to be supported	-	1	Est. total AEC expert time (days)	-	nn
Expected IFC Model Impact (1 *(min)* to 5)	-	1	Est. total Info Modeling expert time (days)	-	nn
Degree of technical difficulty (1 *(min)* to 5)	-	1	Est. total Software/PM expert time (days)	-	nn

2.12.4 Resources Required / Committed

Member Company Resources	**Reqired Days**	**Market Value**		**Days Committed**	**Resource shortfall**
Requirements definition					
Process Model	2	$1000		2	0
Usage Scenaria	4	$2000		4	0
Model design					
Object Model development *(w/ tech.Support)*	10	$5000		10	0
Integration *(w/ tech.Support)*	6	$3000		6	0
Design and Implementation validation					
Test Case development	6	$3000		6	0
Review/feedback on implementations	8	$4000		8	0
Project Management					
Project management and administration	8	$4000		8	0
Travel and Meetings	50	$25000		50	0
Total Member Company Resources	**94**	**$47000**		86	

Model/Specification development support	**Required Days**	**Market Value**		**Days Committed**	**Resource shortfall**
Technical support	nn	$nn			
Project management	nn	$nn			
Publication and Administration	nn	$nn			
Equipment and software	nn	$nn			

Travel and subsistence	nn	$nn			
Total Project Support	**nn**	**$nn**			
Total for Project	**nn**	**$nn**			

Facilities Management

2.13 FM-1 Engineering Maintenance

{{ This project has been delayed for inclusion in Release 3.0 }}

2.14 FM-2 Architectural Maintenance

2.14.1 Project Description

Architectural maintenance is a part of the FM operation and it is concerned with the long term maintainance of the materials and building components (windows, doors etc). Much of the information required is already available within the project model. Additional information is required to establish maintenance operations and to enable the definition for the lifetime of the product.
Architectural maintenance will also provide relevant data to further project work carried out during the building lifecycle. Materials and components have very different life expectancy and consequently they have to be renewed or varied during the lifecycle to suit the operational needs of the building. Provision of relevant and up to date information can improve the knowledge of project participants carrying out the maintenance work and make long term budget for the building

2.14.2 Project Team

Project Leader à Arto Kiviniemi - Nordic

Chapter	*Name*	*Email*	*Hrs / Week*
Nordic	Arto Kiviniemi	Arto.Kiviniemi@vtt.fi	6
Nordic	other		5

2.14.3 Scope of Work

AEC processes to be supported	-	2?	Est. total AEC expert time (days)	-	nn
Expected IFC Model Impact (1 *(min)* to 5)	-	2	Est. total Info Modeling expert time (days)	-	nn
Degree of technical difficulty (1 *(min)* to 5)	-	2	Est. total Software/PM expert time (days)	-	nn

2.14.4 Resources Required / Committed

Member Company Resources	**Reqired Days**	**Market Value**		**Days Committed**	**Resource shortfall**
Process Model	nn	$nn		nn	nn
Usage Requirements	nn	$nn		nn	nn

Object Model development	nn	$nn		nn	nn
Integration	nn	$nn		nn	nn
Test Case development	nn	$nn		nn	nn
Implementation technical support	nn	$nn		nn	nn
Management and Review	nn	$nn		nn	nn
Total Member Company Resources	**nn**	**$nn**		nn	nn

Model/Specification development support	**Reqired Days**	**Market Value**		**Days Committed**	**Support shortfall**
Technical support	nn	$nn			
Project management	nn	$nn			
Publication and Administration	nn	$nn			
Equipment and software	nn	$nn			
Travel and subsistence	nn	$nn			
Total Project Support	**nn**	**$nn**			
Total for Project	**nn**	**$nn**			

2.15 FM-3 Property Management (from the Owner´s viewpoint)

2.15.1 Project Description

Property management is a process starting from requirement programming and continuing through the building’s life cycle. In this phase the FM-3 project covers just a subset of this process focusing on grouping of spaces and other possible objects for different purposes, like maintenance, administration, public registers, mapping etc. This process is based on objects provided by the design and construction process and uses mainly the attributes in the current model. The main user is the building owner and the benefit is more efficient use of the building data and through this cost savings in the administrative work. This process starts after the building is completed and is carried out through the whole life cycle of the building.

2.15.2 Project Team

Project Leader à Poul Sorgenfri Ottosen - Nordic Chapter

Chapter	*Name*	*Email*	*Hrs / Wk*
Nordic	Poul Sorgenfri Ottosen	pso@aua.auc.dk	10
Nordic	Jan Karlshøj	jakbyg@carlbro.dk	3
Nordic	Arto Kiviniemi	arto.kiviniemi@vtt.fi	5
NA	? Kevin Yu		?
Singapore	? Tan Kee Wee		?
UK	? Mike Goodman		?
Total for project			

2.15.3 Scope of Work

AEC processes to be supported	-	3	Est. total AEC expert time (days)	-	30
Expected IFC Model Impact (1 *(min)* to 5)	-	1	Est. total Info Modeling expert time (days)	-	15

Degree of technical difficulty (1 *(min)* to 5) - 1 Est. total Software/PM expert time (days) - 10

2.15.4 Resources Required / Committed

Member Company Resources	**Reqired Days**	**Market Value**		**Days Committed**	**Resource shortfall**
Requirements definition					
Process Model	10	$5 000		10	0
Usage Scenaria	20	$10 000		20	0
Model design					
Object Model development *(w/ tech.Support)*	6	$3 000		?	?
Integration *(w/ tech.Support)*	9	$4 500		?	?
Design and Implementation validation					
Test Case development	5	$2 500		?	?
Review/feedback on implementations	5	$2 500		?	?
Project Management					
Project management and administration	10	$5 000		10	0
Travel and Meetings	10	$10 000		10	0
Total Member Company Resources	**75**	**$42 500**		50 + ?	?

Model/Specification development support	**Required Days**	**Market Value**		**Days Committed**	**Resource shortfall**
Technical support	nn	$nn			
Project management	nn	$nn			
Publication and Administration	nn	$nn			
Equipment and software	nn	$nn			
Travel and subsistence	nn	$nn			
Total Project Support	**nn**	**$nn**			
Total for Project	**nn**	**$nn**			

2.16 FM-4 Occupancy Planning (incl. Design and layout of workstations)

2.16.1 Project Description

This project includes the following three processes:

- Occupancy Planning
- Design of Workstations
- Layout of Workstations for an Open Office

The occupancy planner (includes interior designers, facilities managers, architects, furniture dealers' designers, etc.) applies standards during the assignment of people and organizations to interior spaces. It also involves the planning and moving of building assets such as equipment and furniture. This process occurs during the initial planning of space occupancy, and whenever that occupancy needs to change (company reorganization, company growth, or new hires, etc.). The layout and design of typical workstations can be sub-processes of the occupancy planning when it involves systems furniture planning for open offices. These processes require information about the building floor spaces. They also generate space occupancy data for future use of office planning.

Automatic input and utilization of the IFC supported object data, such as building elements and spaces as well as FF&E and occupants, may improve the efficiency of the processes. New objects generated will also be IFC compliant so that they can be used by varies FM processes during the operation of the facility.

2.16.2 Project Team

Project Leader à Kevin Yu - NA

Chapter	Name	Email	Hrs / Wk
NA	Rick Bartling / Karen Smith-Hosner	rbartling@hermanmiller.com ksmithhosner@hermanmiller.com	3.5
NA	Vicky Borchers	vicky@mksinfo.qc.ca	7
NA	Rolanda Derderian	rolanda@meritt.com	3.5
NA	Francois Grobler	f-grobler@cecer.army.mil	7
NA	Chia Y. Han/ Carl Ruther	chia.han@uc.edu	4
NA	Kevin Yu	kevin@naoki.ca	12.5
NA	other (e.g. IBM, IFMA/CAC, etc.)		?
UK	Paul Chadwick	fax: 117-943-4113	?
Germany	???	???	?
			37.5
Total for project			

2.16.3 Scope of Work

AEC processes to be supported	-	3	Est. total AEC expert time (days)	-	29
Expected IFC Model Impact (1 *(min)* to 5)	-	5	Est. total Info Modeling expert time (days)	-	61.5
Degree of technical difficulty (1 *(min)* to 5)	-	4	Est. total Software/PM expert time (days)	-	32

2.16.4 Resources Required / Committed

Member Company Resources	**Reqired Days**	**Market Value**		**Days Committed**	**Resource shortfall**
Requirements definition					
Process Model	15	$4,725		15	0
Usage Scenaria	15	$4,725		15	0
Model design					
Object Model development *(w/ tech.Support)*	30	$9,450		30	0
Integration *(w/ tech.Support)*	15	$4,725		15	0
Design and Implementation validation					
Test Case development	25	$7,875		15	10
Review/feedback on implementations	7.5	$2,363		0	7.5

Project Management					
Project management and administration	15	$4,725		11	4
Travel and Meetings	60	$4,800		60	0
Total Member Company Resources	**132.5**	**$43,388**		**161**	**21.5**

Model/Specification development support	**Required Days**	**Market Value**		**Days Committed**	**Resource shortfall**
Technical support	nn	$nn			
Project management	nn	$nn			
Publication and Administration	nn	$nn			
Equipment and software	nn	$nn			
Travel and subsistence	nn	$nn			
Total Project Support	**nn**	**$nn**			
Total for Project	**nn**	**$nn**			

Simulation

2.17 SI-1 Visualization

2.17.1 Project Description

In the design of a building or other structure, the architect or designer may want to see what the building or the structure will look like, or may want to render images for the client's benefit. Such visualization may be desired at any time from the earliest architectural design or retrofitting to the final interior design. Visualization is the key to solving lighting and daylighting design problems, and is also important in assessing building performance and human comfort issues. IFC support of this process may reduce input preparation time by 75-85% process (through automatic acquisition of building geometry and all surface properties) and thus make the use of the corresponding applications economically feasible.

2.17.2 Project Team

Project Leader: Vladimir Bazjanac, North American Chapter

Chapter	*Name*	*Email*	*Hrs / Week*
North American	Vladimir Bazjanac	vlado@gundog.lbl.gov	as needed/possible
U.K.	Sandy Kinghorn	100412.3254@compuserve.com	?

2.17.3 Scope of Work

AEC processes to be supported	- 3	Est. total AEC expert time (days)	- 1
Expected IFC Model Impact (1 *(min)* to 5)	- 1	Est. total Info Modeling expert time (days)	- 1

Degree of technical difficulty (1 *(min)* to 5) - 1 Est. total Software/PM expert time (days) - 1

2.17.4 Resources Required / Committed

Member Company Resources	**Reqired Days**	**Market Value**		**Days Committed**	**Resource shortfall**
Process Model	3	$2,250		0	$2,250
Usage Requirements	1	$750		0	$750
Object Model development	.5	$375		0	$375
Integration	0	$0		0	$0
Test Case development	5	$3,750		0	$3,750
Implementation technical support	0	$0		0	$0
Management and Review	1	$750		0	$750
Total Member Company Resources	**10.5**	**$7,875**		0	**$7,875**

Model/Specification development support	**Reqired Days**	**Market Value**		**Days Committed**	**Support shortfall**
Technical support	0	$0			
Project management	0	$0			
Publication and Administration	0	$0			
Equipment and software	0	$0			
Travel and subsistence	0	$0			
Total Project Support	**0**	**$0**			
Total for Project	**10.5**	**$7,875**			

Structural Engineering

2.18 ST-1 Steel Frame Structures

{{ Project Summary for this project not yet available }}

2.19 ST-2 Concrete Frame Structures

{{ This project has been delayed for inclusion in Release 3.0 }}

2.20 ST-3 Sub-Structure Design

{{ This project has been delayed for inclusion in Release 3.0 }}

2.21 ST-4 Structural Loads Definition

{{ Project Summary for this project not yet available }}

Cross Domain Projects

2.22 XM-1 Referencing External Libraries

{{ This project has been delayed for inclusion in Release 3.0 }}

2.23 XM-2 Project Document Management

2.23.1 Project Description

Project Document Management refers to all information pertaining to the documents used to estimate, bid, purchase, and manage the building process as well as for use within the Facilities Management domain. This data identifies the document, the author of the document, changes to the document since the last change, and relationships to other documents.

It is being suggested to the group that the first concentration of our work will be on the Contract Drawings represented in the model. It is acknowledged that this is only a small subset of the related documents of the model. We will continue to review the areas affected and complete a framework for our section of work with a complete understanding of what will be reflected in the first pass of our work into the model by the end of our first full meeting to be held at the end of January.

- ***Who performs this process?***
  All software vendors that use drawings, specifications, and sketches during the life cycle of a project. This would include (the Architect's use of) CAD, estimating, scheduling, management, and facilities management software vendors.

- ***When in the project lifecycle it is performed?***
  From the very inception of the project, where these documents are used to define the project, through the construction of the project with all of its changes, through the management of the "building" once the project is complete.

- ***What other processes does it relate to (input from/output to/controlled by)?***
  This process starts in the creation and modification of the documents and outputs to all processes that use the documents as a means of identification. This would include estimating where changes to the work are usually quantified by document, management, where the documents are used to control the flow of work on a project and establish what is being built by document, and Facilities Management, where documents are the prime method of identifying actual conditions in a facility.

- ***What is the benefit (time or cost savings) in IFC based application support of these processes?***

The control of the project over time depends upon the comparison of many baselines of data from one point in time to another. These baselines are reflected as (can be seen as) documents with a reflection in time. Without the identification and use of these documents, such as a Change Estimate, applications would not be able to identify themselves as distinct from others. In this way, applications such as Estimating, Purchasing, Scheduling, and Management packages are enabled to provide these standard views of a project model. In addition, where documents are still being used as the preferred method of delivery of information regarding a project, such as various government agencies requiring drawings and members of the project team who are not CAD enabled.

2.23.2 Project Team

Project Leader à Raymond H. Brungard - North American

Please note that the team makeup for this work will be international and cross domain in nature. There are a number of individuals who are interested in this work and I am at this time arranging for the final team size and makeup, without the undue disruption of other groups. It is my intention to make sure that the project team includes members from the CAD and Architectural backgrounds to round out the view of Contract Documents.

Chapter	*Name*	*Email*	*Hrs / Week*
NA	Raymond H. Brungard	rbrungard@tcco.com	7
UK	Graham Storer	G_Storer@tel-consult.co.uk	7
UK	To be named later		4
NA	Ken Herold (part time)	iaiexec	1
	As yet Named CAD Software		7
Nordic	Arto Kiminieri	arto.kiminieri@vtt.fi	7
NA	Mike Cole (part time)		.5
Total for Project			33.5

2.23.3 Scope of Work

AEC processes to be supported	-most	Est. total AEC expert time (days)	- 50
Expected IFC Model Impact (1 *(min)* to 5)	- 2	Est. total Info Modeling expert time (days)	- 5
Degree of technical difficulty (1 *(min)* to 5)	- 4	Est. total Software/PM expert time (days)	- 15

2.23.4 Resources Required / Committed

Member Company Resources	**Required Days**	**Market Value**		**Days Committed**	**Resource shortfall**
Requirements definition					
Process Model	25	$1,250		25	$1,250
Usage Scenaria	25	$1,250		25	$1,250
Model design					
Object Model development *(w/ tech.Support)*	5	$250		5	$250
Integration *(w/ tech.Support)*	5	$250		5	$250
Design and Implementation validation					

Test Case development	10	$500		nn	nn
Review/feedback on implementations	5	$250		nn	nn
Project Management					
Project management and administration	5	$250		nn	nn
Travel and Meetings		$12,000		n/a	nn
Total Member Company Resources	**80**	**$16,000**		nn	nn

Model/Specification development support	**Required Days**	**Market Value**		**Days Committed**	**Support shortfall**
Technical support	3	$200			
Project management	5	$340			
Publication and Administration	5	$340			
Equipment and software	2	$130			
Travel and subsistence	5	$340			
Total Project Support	**20**	**$1,350**			
Total for Project		**$nn**			

2.24 XM-3 IFC Model - Enabling Mechanisms

{{ Project Summary for this project not yet available }}

*** Instructions for Completing Project Summary ***

Writing "Project Summary" documents

This document is provided as a formatting template for preparation of a project summary for IFC Release 2.0. It is structured so that it can be combined with all other project summaries as a key section of the IFC R2.0 Project Plan. Draft 2 of this project plan will be presented to the International Council early in March and a final version of the plan will be summitted for their approval prior to the International meetings in Tokyo (April).

The document is structured so that several process summaries can be combined into a single integrated summary document for all R2.0 projects.

Therefore ***please do not modify the document structure***.

Document structure is roughly:

Heading 1	1.	Project Document TYPE header
Heading 2		Industry Domain
Heading 3	1.1	Project header
Heading 4	1.1.1	Description, Team, Scope, Resouces

Notes:

1. Make a copy of this file for your project and name it using the project ID in the R1prj_5p.xls spreadsheet → using the following pattern → R2ps_XXn.doc (where XXn is the Proj ID in the first column of the spreadsheet). Example: "Completion of the Architectural Model" has an ID of AR-1 --- thus the filename will be "R2ps_AR1.doc".
2. Note that anything enclosed in double curly braces → {{ xxx }} ← are instructions to you (the writer). You must replace these (including the braces) with content as described in these instructions.
3. A example project summary is available on the IAI FTP site at: →\\IAI_Projects\Release_2.0\R2ps_EX1.doc.
4. Please be sure to identify and call your project team members to identify the resource your currently have commitment. This is the most important thing you will do in the next week. If you cannot find enough resource to complete you project within the outline schedule, your project will have to be reduced in scope of delayed until R3.0. Resource requirements to complete your project will be assessed both by you and your team AND by the International Technical Management team (ITM). You are best equipped to estimate the industry expert time to complete the process, usage and model validation (test case) documentation (as in R1.0). The ITM team is best equipped to estimate the technical resource (info modeling and software) that will be required to complete your project.
5. This template extends previous versions of the template for this document in multiple ways, including:
   - *AEC Domain groupings have been added as "Heading 2" which demotes previous headings 2 and 3*

If you have questions please send email!

Regards,

Richard See

2.25 Project Summary Template {{Proj ID}} - {{ Project Name }}

2.25.1 Project Description

{{ provide a brief (1-2 paragraph) description of this project. You may want to include the following types of information:

- *AEC processes to be supported by IFC model after project completion*
- *who performs these processes*
- *When in the project lifecycle are they performed*
- *What is the benefit (time or cost savings) in IFC based application support of these processes }}*

2.25.2 Project Team

Project Leader à {{ your name here }} - {{ your chapter here }}

Chapter	*Name*	*Email*	*Hrs / Wk*
Total for project			

2.25.3 Scope of Work

AEC processes to be supported	- nn	Est. total AEC expert time (days)	- nn
Expected IFC Model Impact (1 *(min)* to 5)	- nn	Est. total Info Modeling expert time (days)	- nn
Degree of technical difficulty (1 *(min)* to 5)	- nn	Est. total Software/PM expert time (days)	- nn

2.25.4 Resources Required / Committed

Member Company Resources	**Reqired Days**	**Market Value**		**Days Committed**	**Resource shortfall**
Requirements definition					
Process Model	nn	$nn		nn	nn
Usage Scenaria	nn	$nn		nn	nn
Model design					
Object Model development *(w/ tech.Support)*	nn	$nn		nn	nn
Integration *(w/ tech.Support)*	nn	$nn		nn	nn
Design and Implementation validation					
Test Case development	nn	$nn		nn	nn
Review/feedback on implementations	nn	$nn		nn	nn

Project Management					
Project management and administration	nn	$nn		nn	nn
Travel and Meetings	nn	$nn		nn	nn
Total Member Company Resources	**nn**	**$nn**		nn	nn

Model/Specification development support	**Required Days**	**Market Value**		**Days Committed**	**Resource shortfall**
Technical support	nn	$nn			
Project management	nn	$nn			
Publication and Administration	nn	$nn			
Equipment and software	nn	$nn			
Travel and subsistence	nn	$nn			
Total Project Support	**nn**	**$nn**			
Total for Project	**nn**	**$nn**			

3. AEC Process Definitions

This section defines the AEC processes that will be enabled in IFC Release 2.0. It includes detailed descriptions of the potential software application functionality that this release of IFC is intended to support.

This section is subdivided into categories based on the industry domain groups represented by the IAI. Within each domain's section are one or many processes that the domain is enabling. For each process we define:

- Overview - should give the read a general understanding of the process and how it relates to other processes
- Scope - what is included and what is not
- Definitions - term used in this process
- Process Diagram - This section consists of a TQM process diagram which defines the flow of information within the process.

Then, for each process task shown in the process diagram, we present:

- Task Description - thorough description of the task
- Usage Scenario - an example usage scenario for completing this process using data, graphics, etc. from a real AEC project

Architecture

3.1 AR-1 Completion of the Architecture Model

The domain processes and usage scenarios included in the IFC 2.0 Release for Architecture include

- Core Design
- 1A. Stair Design
- 1B. Restroom Design
- Roof Design
- Shell Design
- Block & Stack

3.1.1 Building Shell Design

The architect balances the building massing with the elevation aesthetics while performing exterior shell design. Both processes (massing and shell design) evolve and cycle back and forth as each may change aspects of the other. The exterior shell design involves making the massing interesting while using glass fenestration, cladding materials, and details in adornment that create a scale and design motif. Other aspects of this process, that are balanced, are the need for visual access and illumination of the spaces behind the shell, and the issues of attaching and waterproofing the shell. The shell design starts typically after a preliminary space layout and during the building massing studies.

3.1.1.1 Introduction

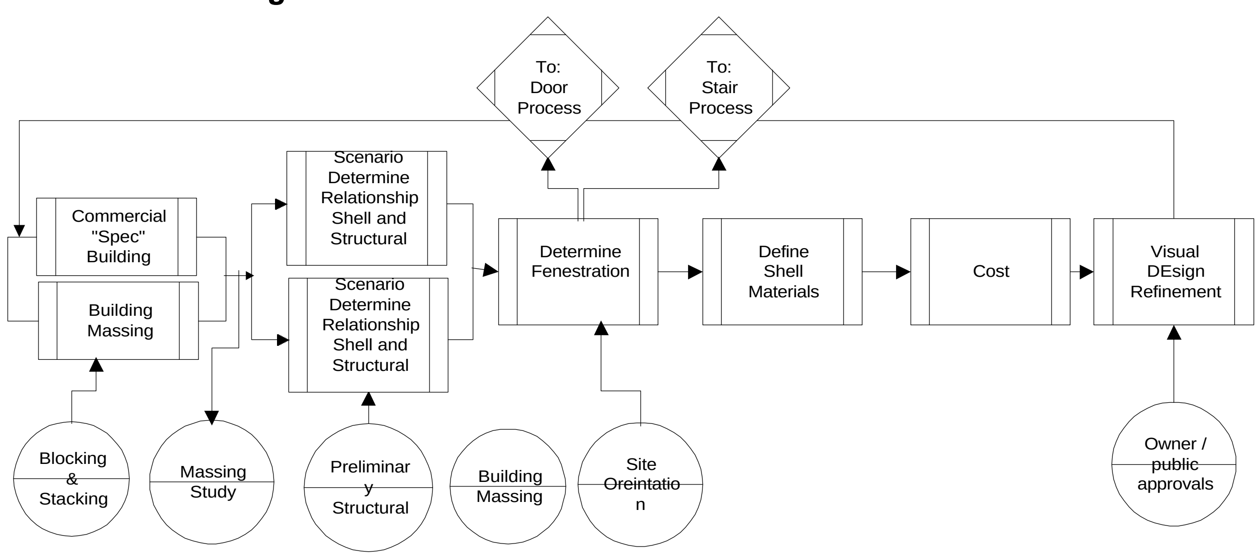
The architect starts the shell design by working with the preliminary stacking and blocking diagrams to determine a massing of the building, based on the floor plates created in the space layout phase. After the massing, the architect will determine the proper aesthetics effect for the building, whether the facade is connected to the outside of the structure or integrated within the structure. The fenestration is determined based on the amount of light and visual impact of the glass and openings on the facade. After the designer determines the type of materials used, preliminary heat gain/heat loss can be calculated for operational cost impact of the building shell. With the final selection of material and fenestration, a detailed design of the adornment of the facade proceeds using reveals, treatment of the materials, cornices, and other building design elements.

***Out-of-Scope**: block and stacking, site analysis and location of the building.*

Definitions:

- *Shell - The exterior wall of a building. Other terms used (facade, elevation, building envelope)*
- *Massing - The exterior shape of a building. A volumetric view of the building.*

3.1.1.2 Process Diagram



3.1.1.3 Usage Scenario

Preliminary Building Massing: The preliminary building massing is a process that is the definition of the volume of the building shape. The massing may be constrained by regional height restrictions and open area standards which are to balance the open area on a site compared to the building footprint area. The massing will also be driven by considering the size of each floor based on a preliminary block and stacking. Client requirement such as optimizing the amount of the occupational space against the exterior wall or the number of corner offices may suggest a shape to the designer. Other subjective issues such as a desire to step the building down to a human scale may drive the massing and shape of the exterior envelope of the building. The floor to floor height of the interior spaces required by the program has a vertical impact on the massing. At this point in the process the designer will start to think about a preliminary structural grid based on a design.

Determine the Relationship between Shell and Structure:

The relationship of the shell and structure is based on the effect the architect wants to achieve with the design. For example, the shell may be attached to an edge of slab and column so the shell hangs and covers the structure. On the other hand, the designer may desire to express the structure and allow the columns and floor slabs to protrude past the shell, in effect using the structure to frame the shell areas. Other design scenarios such as using the structure to shade glass areas may suggest to the designer to extend the structure past the shell.

Determine Fenestration: The fenestration is the design and placement of glass area on the shell to permit natural lighting of building spaces and views from the building. The fenestration is based on the rhythm and aesthetics effect the facade should have with respect to glass area. At this stage, a decision on the shape and size of windows are made but not detailed. The amount of glass area may be driven by the energy criteria and regional location and climate. Each facade or elevation of the shell may have a different fenestration due to the orientation of each building face compared to the direction of the sun during different seasons.

Define Shell Materials: The selection of the shell material is based on a diverse set of criteria. The material may be picked based on the need to fit into other buildings in the area or a regional style or culture. The climate may drive the material selection process along with desires by the client to achieve a style for the building. The durability may create a narrower palate of material. There are also regional construction methods, ease of use, cost, and availability of certain materials that would affect its selection.

Costs: A preliminary analysis may be run to determine the effect of the shell design on the construction and operational cost of the building. The upkeep on the materials along with the construction cost drive the overall life cycle cost of the shell. On the operational side of the equation the quantity and cost of energy to maintain a temperate environment will be determined by the fenestration and materials selected during the design process. Both will have an overall impact on the heat gain and loss of the building shell.

Visual Design Refinements: At this point in the process, the shell is refined and detailed. This may include finishes, additions or treatment to materials such as flame/rough/polished stone, reveals, setting back panels, cornices, or parapets. Each of the adornments, construction techniques, and use of materials are used to apply a character to the design of the facade.

.

3.1.2 Building Core Design

The core design is a balance between making available ancillary spaces and program requirement. The size and location on a floor is determined by the structural systems, program requirements including number of occupants and building codes such as ADA. The design of the core follows the initial layout of the spaces defined in the building program. The spaces that make up the core are typically not defined in the program but are extracted by information about the floor size and occupants.

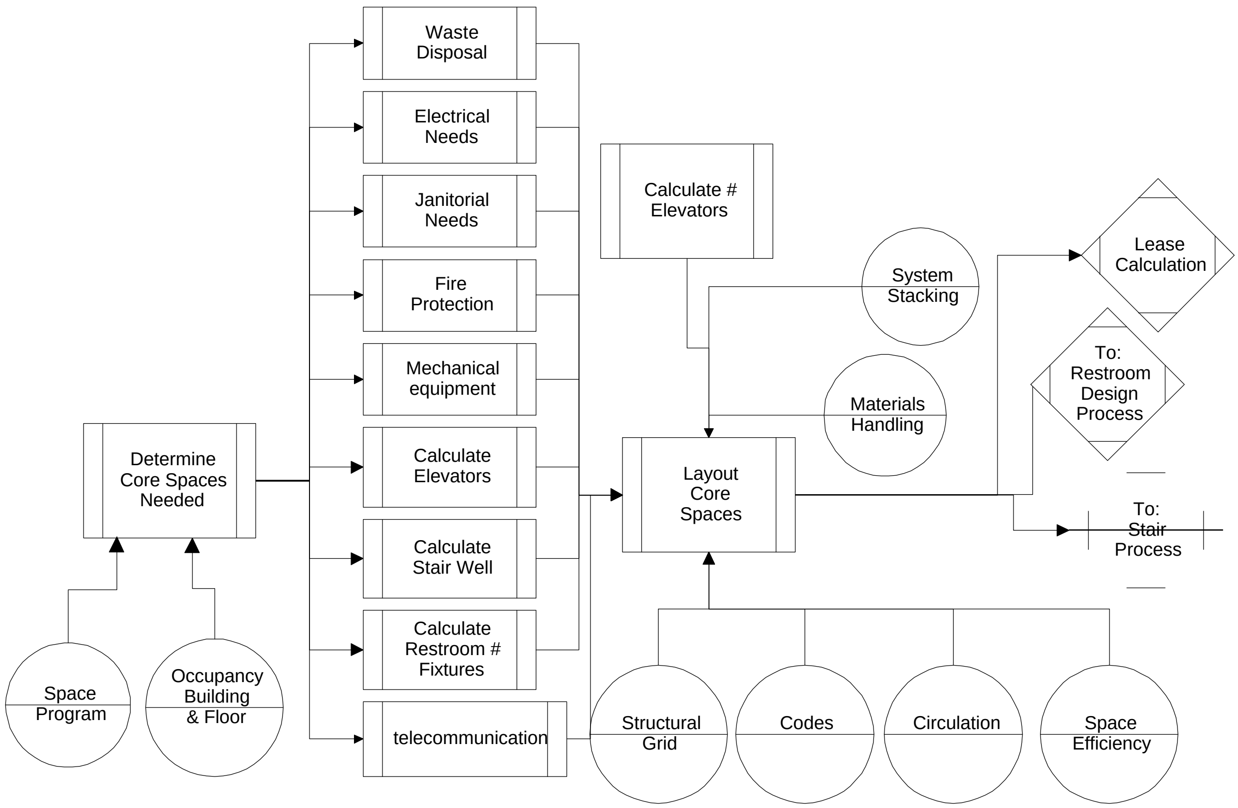
3.1.2.1 Introduction

The core design starts by determining the size of the items needed in the core. Calculations for the number of elevators are based on building occupants and number of floors. The restroom size is based on the number of occupants on the floor and in the building. The floor to floor height is used to determine the length of the stairs which determines the size of the stairwell. The circulation around the core is determined by the type of occupancy and fire codes. The layout of the pieces of the core are driven by the structural grid and distances determined by codes, etc.

Process Scope: Assumptions /presumptions: space program (owners' criteria); occupancy, building, floor; parking garage impacts (structural grids); materials handling (site delivery, building services). The core is defined as items for circulation and service delivery for occupants. It does not have to be in the center of the building.

Out of Scope: This process does not address the actual design of stairs, restrooms, parking design and lobby design. Also materials handling and entering and exiting the building are not included in the core design.

3.1.2.2 Process Diagram



3.1.2.3 Usage Scenario

Determine Core Spaces Needed: The types of core spaces are determined by a range of issues and codes. The floor occupancy, building type, and building codes determine the type and number of spaces needed as part of the core. The types of building services that are needed in the building will determine additional types of spaces to allow passage and access to services central to the buildings operation.

Determine Core Space Sizes: After the determination of which spaces are included in the core for each floor the overall sizes for each needs to calculated. Apply codes and other processes to determine the size and shape of core spaces. The size of service spaces such as chases and shafts are determined by the overall amount of the material such as fluids, gases, and electrical/Telecommunications that have to be passed through and distributed to floors. Spaces used for transporting occupants such as stairs and elevators are calculated based on the volume of circulation determined by the occupancy of the floor and the building they serve. The final areas provided for occupant support such as restrooms are determined by the occupants of each of the floor they reside on.

Layout Core Spaces: The location of the varied spaces in the core is determined by many factors. One of the strongest constraints is the circulation needs for both providing effective space utilization and egress/access to the floor through stairs and elevators. The loads and timing of occupant circulation will determine the number of cabs and ultimately the number of elevator stacks and size of their corresponding shafts. The need to efficiency stacking building services forces the stacking of spaces. The structural needs for sheer walls and the spacing of vertical elements such as columns affects the placement of spaces. If the building includes levels of parking, the trade off between structural bay size and efficient parking layout to optimize the number of parking spaces will affect core element placement.

Detailed Design of Stairs

Covered in this document under Stair design Process.

Detailed Design of Restrooms

Covered in this document under Restroom design Process.

3.1.3 Roof Design

The process of roof design is a mixture of aesthetics, weather dissipation, and hiding other building objects such as telecommunications, mechanical, and elevators. The process is iterative, the designer works back and forth between the massing and roof design to create a building design which expresses a character appropriate to the area, client wishes, and building type.

3.1.3.1 Introduction

The architect determines a type of roof based on the design direction and the character of the building. Using the building massing, the architect lays out the roof. On pitched roofs, refinement of the intersection of the roof planes will be necessary. The architect then determines and designs the drainage. The intersection of the roof with the elevations are designed and detailed. The layout and penetration of other services that are hosted on the roof are considered. Materials are selected.

Definitions:

Dormers (space projection from sloped roof, may be considered standard roof, not unique)

Recreation areas

Helipads

Steeples can also be used as a screen or just ornate

Screening

Chimneys

Vents

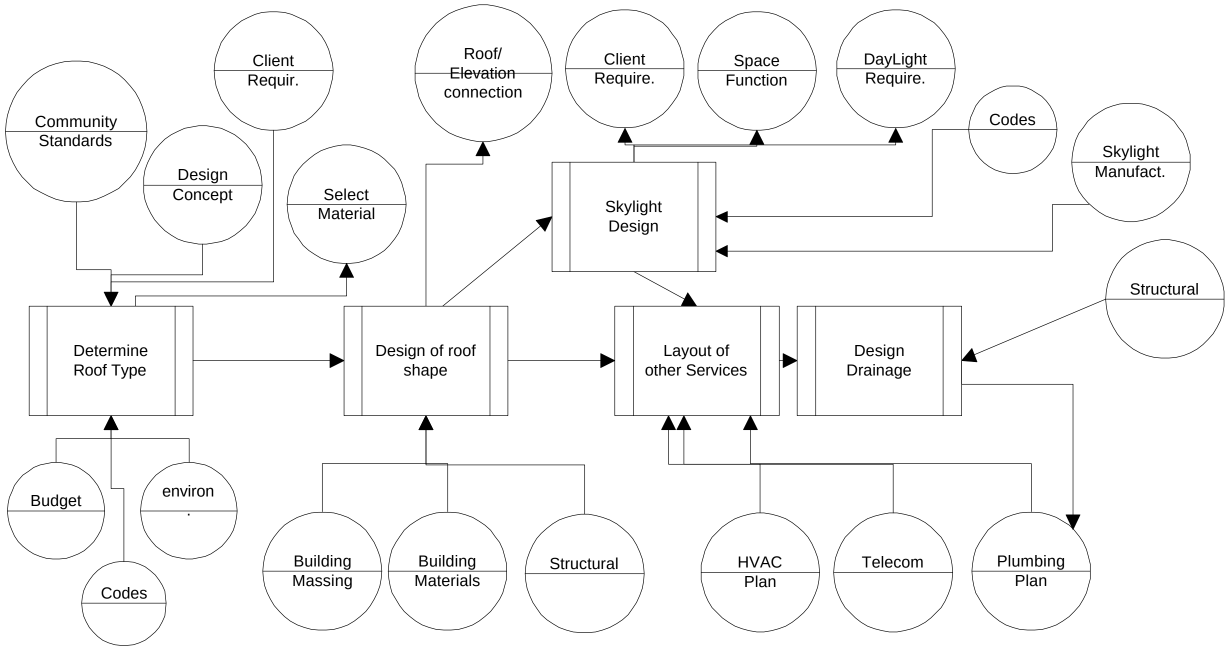
Drainage

Telecommunications: Transmission Tour

Process Scope: Design inputs would cover the process of exterior and interior programs including eaves and overhangs. Interior issues need to address cathedral ceilings, dormers, etc. Exterior roof issues include steeples, parapet roof ventilation, electrical, drainage, recreational areas, planters, irrigation, window washing, skylights, smoke evacuators, access hatches, mechanical screens, roof walk pads, lighting control, and FAA lighting.

Out of Scope: Actual design of electrical, venting, access hatches, smoke evacuators, sidewalk protection canopies.

3.1.3.2 Process Diagram



3.1.3.3 Usage Scenario

Determine Roof Type: The determination of roof type is a balance between form and function of the building. The local style of other building along with the desire of the client for a style effect the final decision. The roof type refers to flat, pitched, gabled, etc. An understanding of the types of services supported by the roof may determine the type of roof selected. The regional climate may dictate a shape of the roof structure to support the amount of wind, precipitation, snow, and also radiation of heat from the sun.

Design Roof Shapes: After the selection of the roof type, a preliminary design is produced to determine the actual shape and its impact on the building form. The slopes of the roof elements to provide the correct sheding of the climatic element will determine pitches. The changes in the massing elements will force the roof to change as new building masses intersect each other. The function of the spaces below the roof may determine the shape along with the need to enclose building services. The type of material used will have a direct impact into the shape of the roof depending on the material constructability. Finally the surrounding building roof-scapes may dictate a direction for the shape.

Skylight/Clear Story: After the shape is created, the integration of any skylights or clear story windows will be integrated into the roof to evaluate the impact and location based on preliminary structural needs. A skylight may not be as simple as a pre-manufactured domed square skylight but could be a complicated barrel vault that runs the length of the building. The intersection of the skylight with the roof becomes critical and may force certain decisions on pitches of roof plains to direct the outside elements away from the glass area.

Layout of Services: With the major roof shape determined and items such as skylights, etc. placed, the designer then looks at the projections through the roof of items such as vents, stair/elevator, telecommunications, glass cleaning, and mechanical. Depending on the size of the projections techniques such as providing screens and other methods to hide the services may be required. Depending on the building program areas such as heliports, health and fitness, and walkways my be required to be included in the roof design.

Design Rain/Snow Drainage: At this point, after the building services are located the sheeding of water needs to be addressed. The runoff of water is calculated based on the roof planes and slopes and a design concept is created to use roof drains, scuppers, or gutters to empty the water from the roof..

3.1.4 Blocking and Stacking

3.1.4.1 Introduction

Overview: Space (area or volume) blocking and stacking is a process of converting the organizational needs of a client into a graphic description of location of spaces and their relationships. After the areas/volumes (bubble diagrams) are created, the designer places them horizontally(blocking) and vertically (stacking).

Process Scope: The scope of this process encompasses: defining spaces, naming of areas/volumes, calculation of spaces, checking mechanisms (max/min standards, confirming design against criteria), establishing adjacencies, grouping spaces, circulation (basic load factoring, % assignment), defining organizational structures, design criteria (site analysis (criteria), grid office layout, day lighting) and existing conditions. Cost referencing or relative quality of spaces (FM reference) may be included. (Need to coordinate with FM domain group)

Out-of-Scope: This process does not intend to address programming, escape (egress) code analysis, or budgeting.

Definitions: Blocking - horizontal placement of spaces

Stacking- vertical placement of spaces

Adjacency - relationship of spaces (both horizontal and vertical)

Space

Grouping, biz-unit, department /relationship to space/adjacency

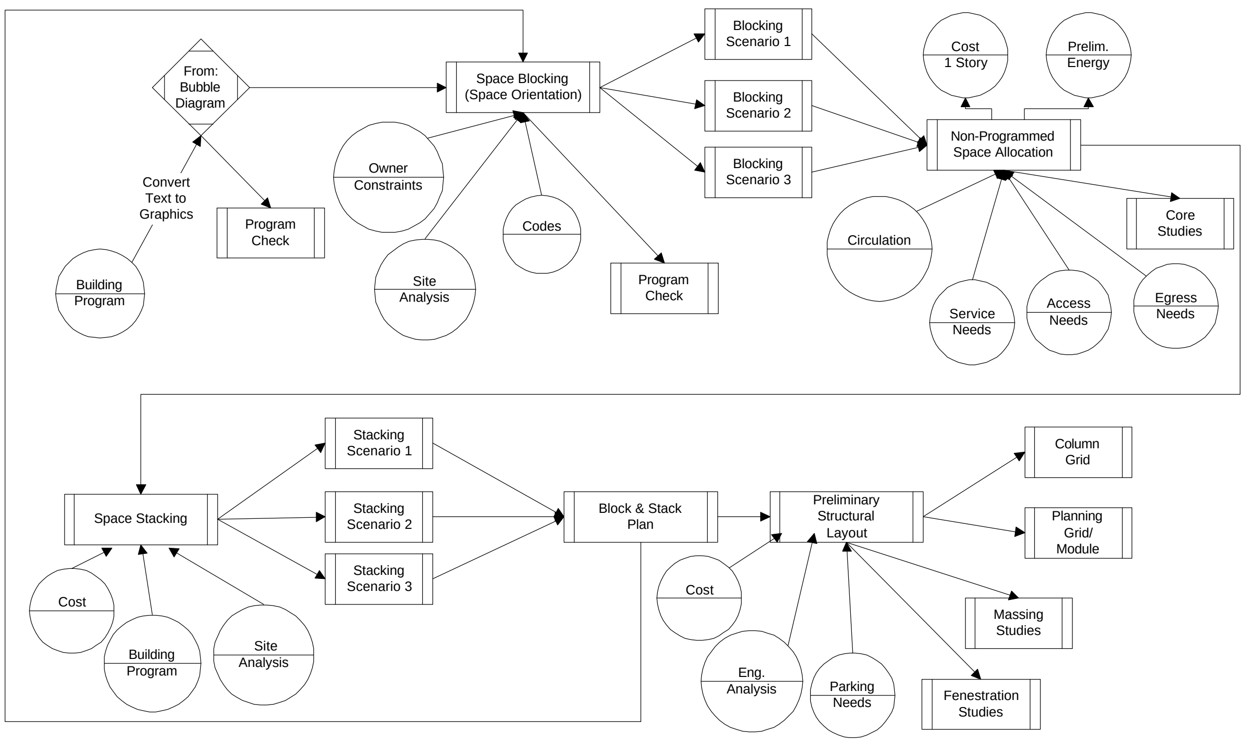
Programmed space

Non-programmed

References: Historical Projects, Project Libraries, Corporate Databases

Contributors: NA, D, UK, No, Sg

3.1.4.2 Process Diagram



3.1.4.3 Usage Scenario

Overview

The blocking and stacking process as an element of conceptual design begins after a building program is defined between the client and architect. The designer starts by creating graphic spaces according to the sizes defined in the building program. While reviewing the adjacency and space size, the spaces are moved around to determine their location horizontally on a floor in the building. The non-programmed spaces such as grouped core elements and circulation are added to the diagram. The process progresses when the vertical location of the space in the building (i.e. stacking is determined). The architect moves between the blocking and stacking tasks until the spaces are organized in an optimal manner. The building structural grid may be refined during this iterative stacking and blocking process.

Subtask Descriptions

Generate Spaces: Generating spaces is a process of converting the alpha and/or numeric information from a building program to an abstract graphic diagram for the beginning of conceptual design. To generate these spaces, the designer will input the building spaces obtained from the program and client. Information about the space such as area, volume, function, adjacency, and occupancy may be entered during this stage. Blocks (or bubbles) depicting the spaces will now be available for conducting a program check.

Space Blocking: Adjacency information will help the designer arrange blocks (bubbles/volumes) in the plan indicating which spaces should be placed next to others. At this point the designer takes into consideration site conditions, codes, owner constraints, municipal constraints, public constraints, and building type constraints to create blocking scenarios. Another check against the building program might also be made after an initial layout.

Non-Programmed Space Allocation: Many programs do not define explicitly the size of circulation spaces and core spaces such as bathrooms, stairs, and elevators. Often, a percentage of occupied space is used to determine the amount of space for circulation or as a result of some other activity. Information such as circulation needs, owner requirements, aesthetic constraints, special use constraints, service and access needs, as well as codes are important at this point of the process. This information along with the designer's insight will help to generate the location of additional spaces not included in the original program. A program check will be performed again.

Space Stacking: For a multi-story building, the designer will stack the grouped spaces vertically on floors. Stacking may involve splitting and moving spaces among floors. The designer uses information used in previous steps to help determine the organization of the floors. Site, code, owner, zoning, public, municipal, and building system constraints aid the designer in stacking spaces. As before, it is also necessary to check this stacking scenario against the building program.

Preliminary Structural Layout: A preliminary indication of structural elements, such as shear walls and column grids, is needed during the process of laying out spaces. The designer might look at parking circulation, various codes, a planning grid, a preliminary structural grid, owner constraints, and constructability at this stage. In addition, the designer will take into account the regional building methods to generate the structural support mechanisms, planning grid/module, and massing of the building. Again, a program check is performed.

3.1.5 Stair Design

Stair design is accomplished by working with the major elements, such as treads, landings, and railings, to determine the appropriate size of the stair and its elements. The process is an iterative process where the answer for one of the elements may change the size of another. The two factors that determine many of the size related decisions are based on the occupancy load and the exiting requirement.

3.1.5.1 Introduction

The architect starts the stair design by working with information about the building such a location of the stair based on egress. The width and depth is defined during a process of working back and forth. The width is determined by the number of occupants traveling through the stairwell during an emergency. The width is typically defined in the local building and fire codes. The floor to floor heights of the story are used to determine the length of the stairs, based on a rise and run. The designer may then design the depth of the landing based on codes. As the design progresses to the handrail, its design can potentially affect the width of the stairs and landing, depending on the distance it protrudes into the stairwell. At the point where the size of the treads, landing, and the handrail are set, the materials and construction methods are determined. The final design involves adding items such as exit signs, doors and hardware, and emergency lighting.

Process Scope: The process described is for fire stairs in a building. Include fire stair materials. ADA safe haven concept should be included (telecommunications, extra design space, area impact)

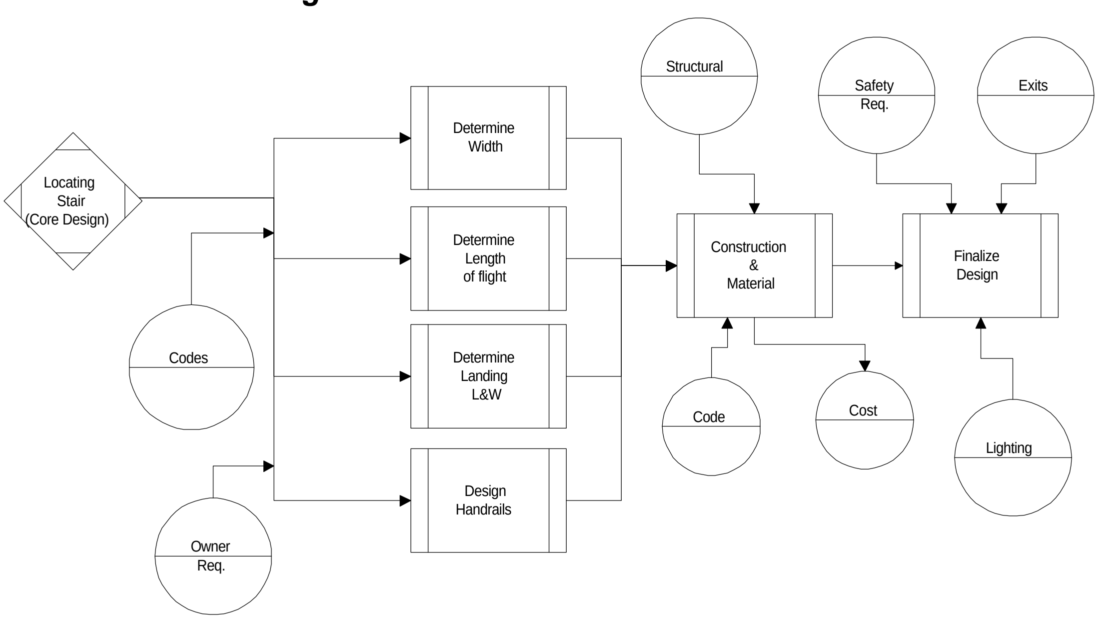
Out of Scope: Ornamental stairs not in scope and not required for exiting a floor, ladders.

Definition: ADA safe haven

References:

- Safe haven documentation
- Calculation of stair rise and run

3.1.5.2 Process Diagram



3.1.5.3 Usage Scenario

Locate Stairs

Covered in this document under the Core design Process.

Determine Width: The width of the stairs are determined by building codes which indicate the minimum sizes based on the number of occupants using the stairwell over a certain amount of time. The designer should take into consideration the depth of the handrail as it protrudes into the stair and cuts down on the actual width of the tread.

Determine Tread and Risers: The length or run of the stairs is dependent on the height between the floors being calculated. There are appropriate height and depth of treads based on what is comfortable for occupants to walk up and down steps without stumbling. The rise/run of the stairs are defined in tables in local building codes.

Determine Landing: The landing performs two functions. First it allows the occupants a place to exit out of a floor onto the stair well. The second function is that it is a location to change directions in the stair well. The landing width and depth is determined by stairs connected to the landing and the number of occupants switching between stair flights. The local building codes describe the appropriate size based on the occupants on each of the floors. A new requirement is the inclusion of a safe haven, which is an alcove on the stair landing where a wheel chair can reside out of the way of stair traffic until help can arrive.

Handrail Design: The handrail has both an aesthetic component and is driven based on codes. When a decision is made on the type of handrail, the width of the stairs may change based on the distance the handrail protrudes from the wall. The height of the handrail from the step to its grab point is also a consideration which is outlined in the local building codes.

Construction and Materials: As the design of the stair is taking shape, a decision on materials is made. The designer selects the material for the stairs such as concrete, steel, or a combination of both. The decision may be based on regional standards, ease of construction, or local fire codes. The materials on the tread and the type and construction of the nosing are also made at this point in the process. The final stage of deciding on the construction the designer determines how the stringer connects the tread, riser, and connects it to the stair well.

Finalize Design: The final detail of stair design evolves other objects connected or part of the stair. This may include deciding on the type of exit doors, signage, standpipe location, location of vents and hatches. Also design of emergency lighting and ventilation should be performed by fire safety engineers at this point in the process.

3.1.6 Restroom Design

The design of restrooms involves effective movement of building occupants, ADA codes, and aesthetic use of materials. The minimum number of fixtures is determined by the number of occupants that reside on a floor or visit a floor.

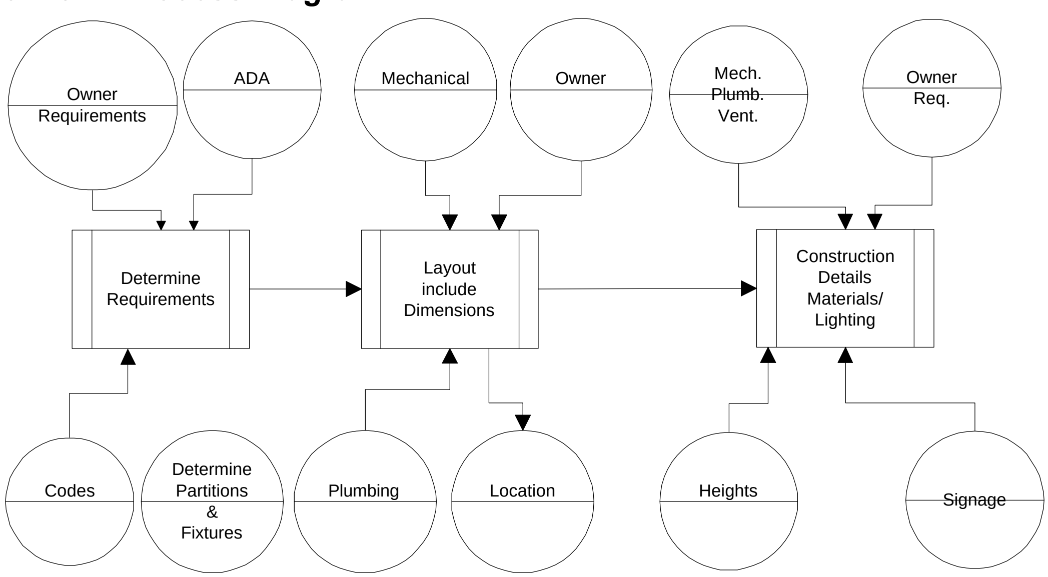
3.1.6.1 Introduction

At the start of restroom design, the number of fixtures are determined by the floor occupancy. The designer will also determine items such as partition type, fixture type, stall sizes, based on codes such as ADA and any client requirements. The next level of design involves locating the restroom fixtures and lavatories to use the most effective amount of space to contain cost but provide effective circulation. The next level of design involves locating the lavatories, mirrors, towel racks, grab bars, hand dryers, and any other object that services the restroom occupants. Appropriate location of fixtures and other items in the restroom may be determined by effective use of other building services such as plumbing stacks, etc. The final step of design is more aesthetic in that it involves the visual character of the restroom in selecting material type, sizes and objects such as faucets etc.

Out-of-Scope: Locker Rooms, Showers

In-scope: Commercial Public Restroom associated with the building core

3.1.6.2 Process Diagram



3.1.6.3 Usage Scenario

Determine Requirements: The number of fixtures which is considered toilets, urinals and sinks is determined by codes and the floor occupancy. The ADA requirements define how many of the fixtures are designed for handicapped access.

Layout: Layout involves the location of the major fixtures and the stalls that surround them while creating appropriate circulation for occupants and handicapped. The effective delivery of services such as water and getting rid of waste will set a common plumbing wall which makes is cost effective by stacking all plumbing services for the building.

Construction Detailing and Finishes and Lighting: Detailed design involves locating other objects inside the restroom such as hand dryers, trash receptacles, outlets, etc. A closer look at other trades, such as Plumbing, HVAC, and Electrical. The final step of the restroom design involves selecting the materials and lighting appropriate for the building type and clients' requirements. The selection of the style of partitions, faucets, and other fixtures such as whether the toilet is wall hanging or rests on the floor is based on the designer's preferences.

3.2 AR-2 Space Planning for Escape Routes

- Means of Escape from Spaces.
- Compartments
- Safe Zones
- Escape Routes

3.2.1 Means of Escape from Spaces

3.2.1.1 Introduction

Overview: Means of escape from an occupied space to a safe space. The process will identify occupancy Use class/classes. compartments, and safe zones. Then the process will define space, openings and adjacent spaces, providing an escape route to safe space.

Process Scope: An itemization of the sub-processes that are within scope for this process.

- Start
- Identify primary occupancies.
- Identify secondary occupancies.
- Identify Use class compartmentation.
- Identify Fire Compartmentation.
- Identify Safe Zones.
- Identify topological route through safe zones.
- Finish

Out-of-Scope:

- Fire Protection to Space enclosure.
- Fire Protection to Elements of structure.
- Fire Protection to Electrical, Mechanical and Plumbing Services.
- Fire Fighting Equipment
- Fire Resistance and Surface Spread of Flame

Definitions:

Space function:	The occupancy of any space is dictated by the functional role of that space.
Use Class Compartment:	Compartment of single Primary (*Principal*) Use with or without Secondary Use
Fire Compartment:	Compartment of single primery Use class or part of a single Use class, with or without related secondary Use classes.
Safe space:	Space separated from Fire compartments. Activity and occupancy within that Safe space does not form a Fire Hazard. Safe space leads to another adjacant safe space, allowing ultimate escape to a final external open Safe space,
Openings:	Clear openings, doors and windows, provide clear and safe access through an enclosure, from one space to another.
Applied finishes:	Finishes applied to floor, walls and ceilings (roofs) and classified under combustible or non- combustible materials.

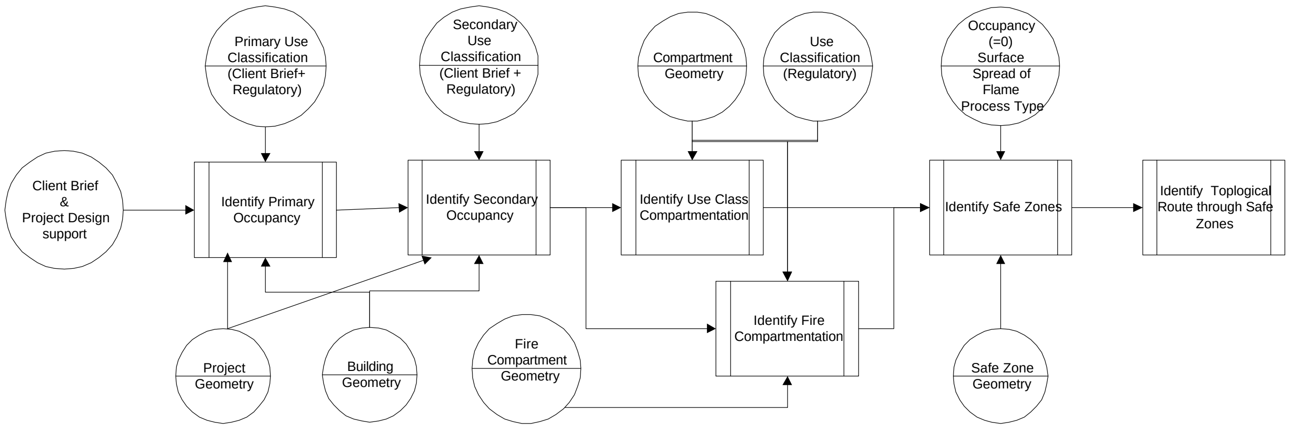
References:

- Uniclass classification.
- Ci/SfB classification - space classification.

Contributors:

- Jay Patankar.
- Steve Race.
- John Cann.

3.2.1.2 Process Diagram



3.2.1.3 Usage Scenario

Overview

On receipt of Architects drawings, identify primary and secondary Use classes, for the total project. Indoing so define shape and size of each Use class. Subject to numbers of Use classes, and their shape & size. Use class compartments will be defined. Use class compartments will be sub-divided into Fire compartments. and Safe (*escape*) zones. The escape routes will be extended through Safe zones, to total safe space, i.e. external space.

Subtask Descriptions

Occupancy use of every building block within the project will be examined; with a view to evaluating its shape, size and distances for each primery Use class. Within the building block there may be other accommodation, which is anciliery to the primery Use, and which is classified as secondary.

Having established all Use classes, subject to shape, size and distances, of each Use class, if necessary separate into individual Fire compartments, to meet the criteria set out by the Fire Authority.

Either within or outside the Fire compartment identify the Safe space, as escape routes.

Safe spaces are Linked and extended to form escape route.

BuildingServices

3.3 BS-1 HVAC Systems Design

The processes being enabled by this domain for this release of IFC's are:

- HVAC System Design
    - HVAC duct design for conveying air
    - HVAC hydronic design for conveying fluids (typically water)

3.3.1 HVAC Duct System Design

3.3.1.1 Introduction

Overview:

Process Scope: Sub-processes that are within scope for this process

- Select and locate air terminal boxes and devices and fan
- Connect components with ducts and fittings
- Locate other system components: dampers, etc.
- Facilitate sizing ducts and fittings
- Facilitate interference checking
- Facilitate pressure loss calculations
- Facilitate fan selection
- Generate final system representation

Out-of-Scope: Sub-processes that have been purposely omitted from this process

- Selection of system type
- Actual sizing the duct and fittings
- Performing interference checks
- Performing pressure loss calculations
- Performing fan selection

Definitions:

- ASHRAE - American Society of Heating Refrigeration and Air Conditioning Engineers
- SMACNA - Sheet Metal and Air Conditioning Contractor's National Association

References:

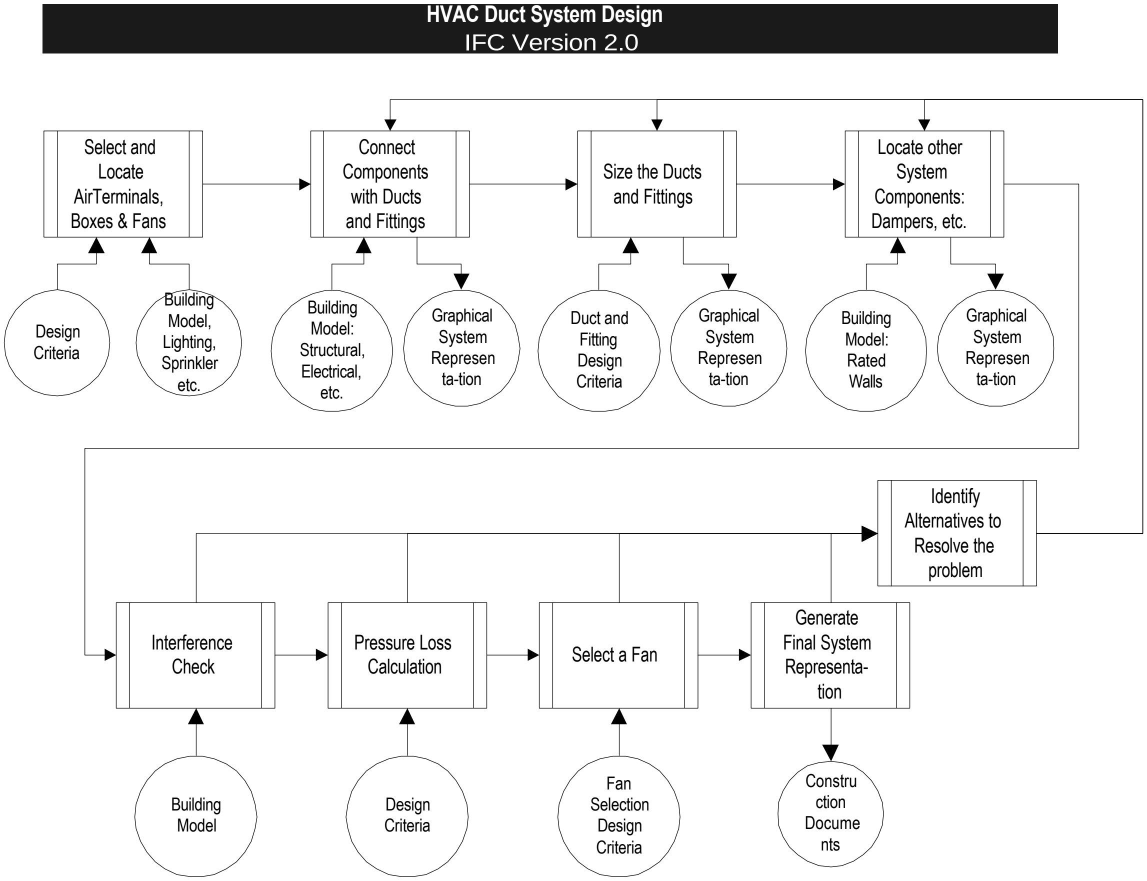
- ASHRAE Handbooks
- SMACNA HVAC Duct Construction Standards

Contributors:

Project Leader à Jim Ahart (Domain) Jim Forester (Technical)

- North American Chapter -- Building Services Committee
- United Kingdom Chapter -- Building Services Committee
- German Chapter -- HVAC Committee
- Nordic Chapter -- HVAC Committee
- French Chapter

3.3.1.2 Process Diagram



3.3.1.3 Usage Scenario

Overview

Once an appropriate system type has been determined (outside of scope), the HVAC Duct System Design process begins by selecting and locating air terminal devices, air terminal boxes and fans that will be part of the system. Reflected ceiling plans may be available showing light fixtures, sprinklers and the ceiling grid to aid in the location of air terminal devices. If these are not available the engineer selects locations for the air terminal devices and submits the locations to other members of the design team for coordination. To appropriately locate the air terminal boxes and devices, a structural drawing is required so that initial interference may be avoided.

The next step is to connect the air terminals, terminal boxes and fans together with ducts and fittings. A graphical representation of this system layout is generated for use in calculating duct sizes and coordination with other disciplines.

The air flow rates are assigned to the air terminals. These air flow rates are determined by the building load calculations, and these processes are defined in the IFC 1.0 Specifications.

The duct and fitting sizes will then be calculated based on these air flow rates and the duct system design criteria. The duct and fitting sizes are then updated in the graphical representation of the system.

Other required system components (i.e., dampers, sensors, etc.) are then located on the graphical representation. This process requires the fire rated walls, exit corridors, etc. are available from the architectural plans. Any components that require other disciplines to respond are identified, such as electrical power required to motorized dampers.

Once these components are located, an interference check is performed. This requires the coordination with the other building disciplines and may require resizing or relocating ducts, fittings, etc.

A final duct system pressure loss calculation may be required beyond that made during the duct sizing based on changes from estimated values to actual values that can only be determined after the duct sizes are finalized. With the final pressure loss, the total air flow and the engineering design criteria, a fan can be selected.

Primary difficulties in the duct system design process are coordination with other disciplines to prevent conflicts for space on the job and to predict sound produced from air flow in the ducts and air terminals.

Select and Locate System Components

Generally there are three levels of equipment to be selected and located.

- **Fan:** The location of the fan used for moving the air in the duct system. The fan may be for supplying, returning or exhausting air from the building or space. The fan may be a stand alone fan or part of a manufactured assembly which may include coils, filters, mixing boxes, etc. Combination fans, coils etc. may be factory assembled or assembled at the project site. The exact size and capacity of the fan are not required at this stage, though an approximate fan size is necessary to ensure the space selected for the fan is adequate. Though not essential, having the size of the fan outlet is useful in sizing the transition between the fan outlet and the duct.
- **Air Terminal Boxes:** Depending on the type of HVAC system, the system may or may not have air terminal boxes. Terminal boxes are typically located in a branch duct from the main supply duct. There are several different types of terminal boxes. They are used in various ways to control the amount and or temperature of the air being supplied to one or more spaces with similar heating and cooling load characteristics. It is desirable but not necessary to know the exact terminal boxes being used in order to size the ducts system. If the exact terminal box being used is known, the exact size duct connections and pressure drop through the terminal boxes are known. Also terminal boxes from different manufactures have different dimensions and knowing the exact dimensions and clearances required for maintenance can prevent future conflicts for space.

  Terminal boxes are typically located after the air terminal devices used for distributing the air in the spaces are located. This allows the terminal boxes to be positioned to permit the shortest duct runs between the terminal box and the air terminal devices it supplies.
- **Air Terminal Devices:** Air terminal devices are used to distribute the air from the duct system to the spaces or to remove air from the spaces. The air terminal device can be to the supply, return or exhaust air ducts in different ways:
  - Directly into the side of a main or branch duct or on a short duct section that allows for a volume damper and/or a lower resistant transition from the duct to the air terminal device. This type of connection is used where the duct is exposed in the space.
  - Directly on the outlet of a terminal box.
  - On the end of a branch duct from the main duct or from a duct on a terminal box. The air terminal device may terminate in an opening in the ceiling, through the wall or be exposed in the space.
  - An air terminal device can simply connect to an opening through a wall into a chase that is part of the building or to a ceiling plenum used for return air. Locating an air terminal devices used in this way are not required for sizing the duct system, they are usually located at the same time other air terminal devices are located.
  - Selecting the exact air terminal devices and their accessories at this stage is not required to size the duct system but is desirable to keep from revisiting each of the air terminal devices a second time. Making a selection also supplies the exact duct connection required and the exact pressure drop through the air terminal device which is necessary in the final design of the duct system for the fan selection.

Connect the Components with Ducts and Fittings

This step involves preparing drawings which will schematically represent the system under design. The duct is typically drawn from the fan to the air terminal boxes, if any, and then to the air terminal devices. There are different types of elbows, tees and other fittings so each fitting must be designated as to what type is being used. These schematics are then used to begin coordination with other disciplines which are impacted by the duct system. The information derived from the air flow associated with each air terminal device and the schematic drawing is used by a duct sizing program.

Sizing the Duct and Fittings

The ducts are sized using the information derived from the schematic drawing and design the criteria established by the engineer. Design criteria include such things as type of design (constant pressure, static regain, etc.), maximum velocity, maximum height of duct, material to be used, etc. The actual sizing of the duct and fittings is out of scope, as this is application specific.

Locate Other System Components

Other components, such as fire dampers, volume control dampers, louvers, filters, etc., are located on the drawing. These components have pressure drops that may only be precisely determined after the actual duct sizes are known. After these pressure drops are determined and added to the input the total duct pressure drop is calculated. In many cases the pressure drop for these components are known before the ducts are sized or can be closely estimated so they can be entered before sizing the ducts and fittings.

Interference Checks

Interference checking can reveal areas where changes are required in the location of specific ducts, or the height of the duct is too great requiring a transition fitting to clear a beam or a pipe. After any interference are corrected the total pressure drops for the system can be calculated. Performing the actual interference check is out of scope, as this is application specific.

Pressure Loss Calculations

After any interference are corrected the total pressure drops for the system can be calculated. Performing the actual pressure loss calculations is out of scope, as this is application specific.

Fan Selection

With the total air flow and pressure requirements established, the engineering design criteria for the fan, such as class, type (forward curve, backward curved, in-line, etc.), material for housing and wheel or blades, etc., maximum sound level are selected. Fans that come as part of preassembled units such as in the typical roof top unit, do not allow for many of these options. After all of the fan criteria are established the actual fan selection is typically made using a fan manufacturer’s fan selection program. Performing the actual fan selection is out of scope, as this is application specific.

Generate Final System Representation

After the components are selected and the duct and fittings sized the results are used to generate drawings showing the actual size and location of the ducts and all of the components.

3.3.2 HVAC Hydronic System Design

3.3.2.1 Introduction

Overview:

Process Scope: Sub-processes that are within scope for this process

- Select and locate equipment to be connected in the hydronic system
- Connect components with pipe and fittings
- Add other Components: strainers, valves, etc.
- Facilitate sizing pipes and fittings
- Facilitate interference checking
- Facilitate pressure loss calculations
- Facilitate pump selection
- Facilitate flow analysis
- Generate final system representation

Out-of-Scope: Sub-processes that have been purposely omitted from this process

- Selection of system type
- Actual sizing the pipe and fittings
- Performing interference checks
- Performing pressure loss calculations
- Performing pump selection
- Performing flow analysis

Definitions:

- ASHRAE - American Society of Heating Refrigeration and Air Conditioning Engineers

References:

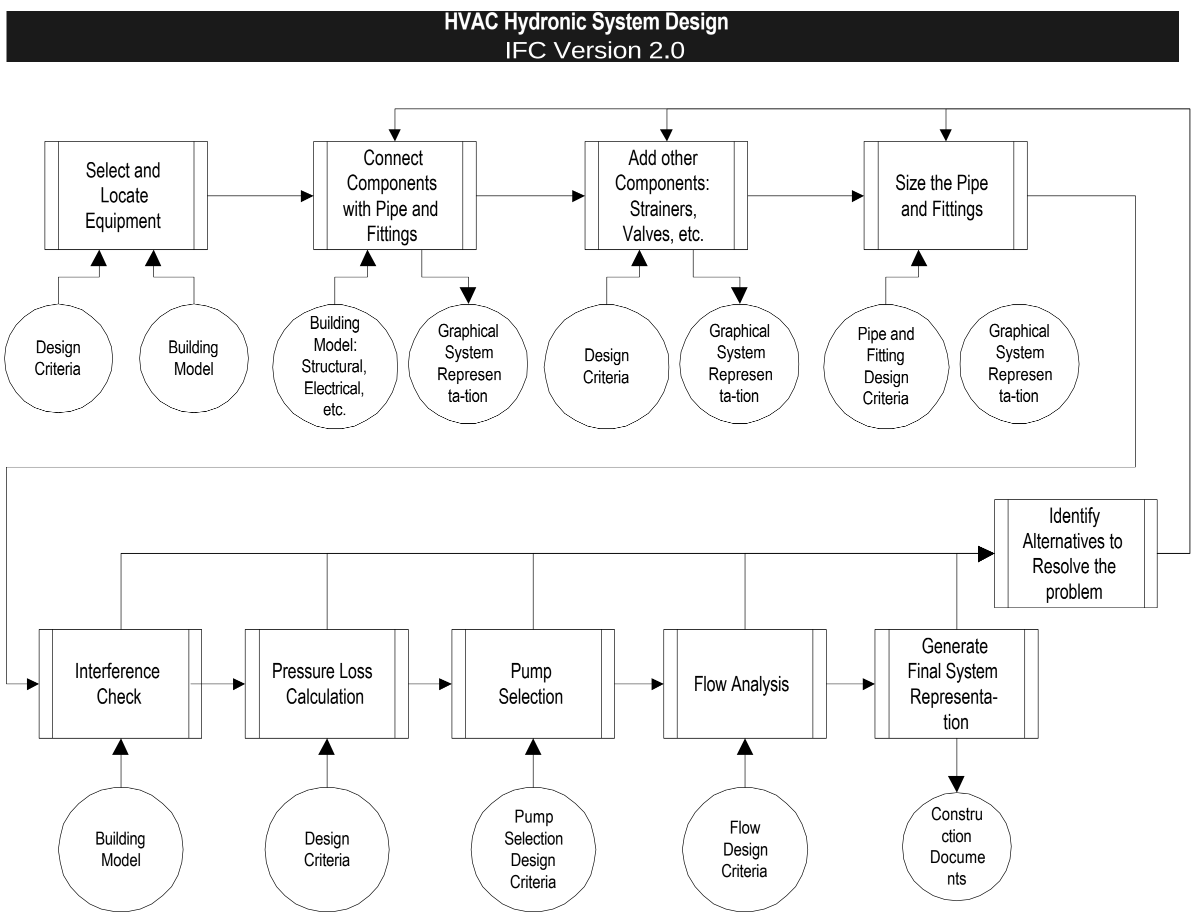
- ASHRAE Handbooks

Contributors:

Project Leader à Jim Ahart (Domain) Jim Forester (Technical)

- North American Chapter -- Building Services Committee
- United Kingdom Chapter -- Building Services Committee
- German Chapter -- HVAC Committee
- Nordic Chapter -- HVAC Committee
- French Chapter

3.3.2.2 Process Diagram



3.3.2.3 Usage Scenario

Overview

Once an appropriate system type (chilled water, hot water, condenser water, etc.) has been determined (outside of scope), the Hydronic System Design process begins by selecting and locating pumps, coils, chillers, boilers, heat exchangers, cooling towers, etc., that will be part of the system. Building floor plans and elevations are required to locate equipment. To appropriately locate the equipment a structural drawing is required so that initial interferences may be avoided.

The next step is to connect the various pieces of equipment together with pipes and fittings, and noting the types of fittings (90 degree elbow, 90 degree long-radius elbow, 45 degree elbow, thru tee, etc.). A graphical representation of this system layout is generated for use in calculating pipe sizes and coordination with other disciplines.

The fluid flow rates and pressure drops are assigned to the coils, heat exchangers, or other pieces of equipment that remove energy from the system. The fluid flow rates are determined by the building load calculations, which processes are defined in the IFC 1.0 Specifications, and the engineer's design criteria.

The pipe and fitting sizes will then be calculated based on these fluid flow rates and the pipe system design criteria. The pipe and fitting sizes are then updated in the graphical representation of the system.

Other required system components, such as valves, strainers, etc., are then located on the graphical representation. Any components that require other disciplines to respond are identified, such as electrical or pneumatic power required to operate control valves.

Once these components are located, an interference check is performed. This requires the coordination with the other building disciplines and may require relocating some pipes.

A pressure loss calculation is made. With the final pressure loss, the total fluid flow and the engineering design criteria, a pump can be selected.

Primary difficulties in the pipe system design process are coordination with other disciplines to prevent conflicts for space on the job and to predict sound produced from rotating equipment and fluid flow in pipes.

Select and Locate System Components

The selection of equipment (coils, evaporators, condensers, unit heater, radiation, etc.) the piping system will serve is made by the designer, using information from the heating and cooling load calculations in conjunction with manufacturers' equipment information and engineering judgment. The selection of equipment should include the type and size of pipe connections to the equipment. The equipment is located on the drawing, giving consideration to space requirements for removal of coils and tube bundles, and servicing for cleaning.

Connect the Components with Pipe and Fittings

This step involves preparing drawings which will schematically represent the system under design. There are different types of elbows, tees and other fittings so each fitting must be designated as to what type is being used and its pressure drop characteristics. These schematic drawings are then used to begin coordination with other disciplines which are impacted by the hydronic system. Interferences must include pipe hangers and supports, and insulation when applicable. Often the piping around a given type of coil, unit heater or other piece of equipment has been standardized in a design office. These assemblies of pipe, valves, fittings, etc., can be treated as macro piping systems related to the given piece of equipment.

Add Other System Components

Other components, such as valves, strainers, etc., are located on the drawing. These components have pressure drops, and connections that may be different from the pipe size. The requirement for some or all of these components may come from equipment selection programs, from standard lists or libraries, or be determined manually by the engineer.

Sizing the Pipe and Fittings

The pipe and fittings are sized using the information derived from the schematic drawing and the design criteria established by the engineer. Design criteria includes, such things as maximum velocity, pipe material to be used, etc. The actual sizing of the pipe and fittings is out of scope, as this is application specific.

Interference Checks

Interference checking can reveal areas where changes are required in the location of pipes. Interference checking must include space for insulation, pipe supports and operating and servicing of valves, strainers, etc. For example, placing valves with stems down is not good engineering practice, while a horizontal stem requires more space for the stem, and a vertical stem may require more space at the side for service access. Performing the actual interference check is out of scope, as this is application specific.

Pressure Loss Calculations

After any interference are corrected the total pressure drops for the system can be calculated. Performing the actual pressure loss calculations is out of scope, as this is application specific.

Pump Selection

With the total fluid flow and pressure requirements established, and the engineering design criteria for the pump, such as in-line, base mounted, materials, etc., are determined the pump selection can be made using a pump manufacturer s pump selection program. With the selection of the pump consideration must be given to isolating the pump from the influence of expansion and contraction of the piping system due to temperature changes, and to the transfer of noise from the pump to the building. Performing the actual pump selection is out of scope, as this is application specific.

Flow Analysis

For hydronic piping systems with diversified loads so that all coils do not need maximum flow at the same time, or where there are multiple pumps in the system, the exact flows, pressure drops and coil performance become unpredictable. Under these conditions good engineering practice requires further analysis of the flow. The results obtained from the pipe sizing program are necessary to the use of a flow analysis program. Performing the flow analysis is out of scope, as this is application specific.

Generate Final System Representation

After the components are selected and the pipe and fittings sized the results are used to generate drawings showing the actual size and location of the pipes, fittings and all of the components.

3.4 BS-2 Power and Lighting Systems Design

{{ Process definition and Usage Scenaria for this project not yet available }}

3.5 BS-3 Pathway Design and Coordination

{{ Process definition and Usage Scenaria for this project not yet available }}

3.6 BS-4 HVAC Loads Calculation

{{ Process definition and Usage Scenaria for this project not yet available }}

Client Briefing

3.7 CB-1 Client Briefing

{{ This Project has been delayed for inclusion in Release 3.0 }}

Construction / Construction Management

3.8 CM-1 Procurement and Logistics

{{ This Project has been delayed for inclusion in Release 3.0 }}

3.9 CM-2 Temporary Construction

{{ This Project has been delayed for inclusion in Release 3.0 }}

Codes and Standards

3.10 CS-1 Code Compliance Enabling Mechanism/ Energy Code Compliance Checking

Processes Defined in this project:

1. Commercial and Residential Energy Code Compliance Checking

3.10.1 Commercial and Residential Energy Code Compliance Checking

3.10.1.1 Introduction

Overview:

This process will support applications that determine whether buildings conform with energy-efficiency codes for new construction. The CS-1 project will focus on two model codes that are widely used in the United States. The project will primarily address requirements pertinent to building envelope and lighting.

Process Scope:

- Commercial energy code compliance (e.g., ASHRAE/IES 90.1-1989 [Code])
- Residential energy code compliance (e.g., MEC)
- Prescriptive code requirements

- Performance code requirements

Out-of-Scope:

- Determination of which codes apply
- Modeling of code requirements (i.e., the object model will not include the code requirements)
- Modeling of energy code provisions not normally addressed on the building plans; e.g. compliance procedures, detailed product and construction specifications, and other information normally relegated to project specifications.

Definitions:

- MEC: Model Energy Code
- HVAC: heating, ventilating, and air-conditioning

References:

- Model Energy Code, The Council of American Building Officials; Falls Church, VA; 1993.
- ASHRAE/IES Standard 90.1-1989, Energy Efficient Design of New Buildings Except Low-Rise Residential Buildings; ASHRAE, Atlanta, GA; 1989.
- Energy Code for Commercial and High-Rise Residential Buildings, Codification of ASHRAE/IES 90.1-1989 Energy Efficient Design of New Buildings Except Low-Rise Residential Buildings; ASHRAE, Atlanta, GA; 1993.

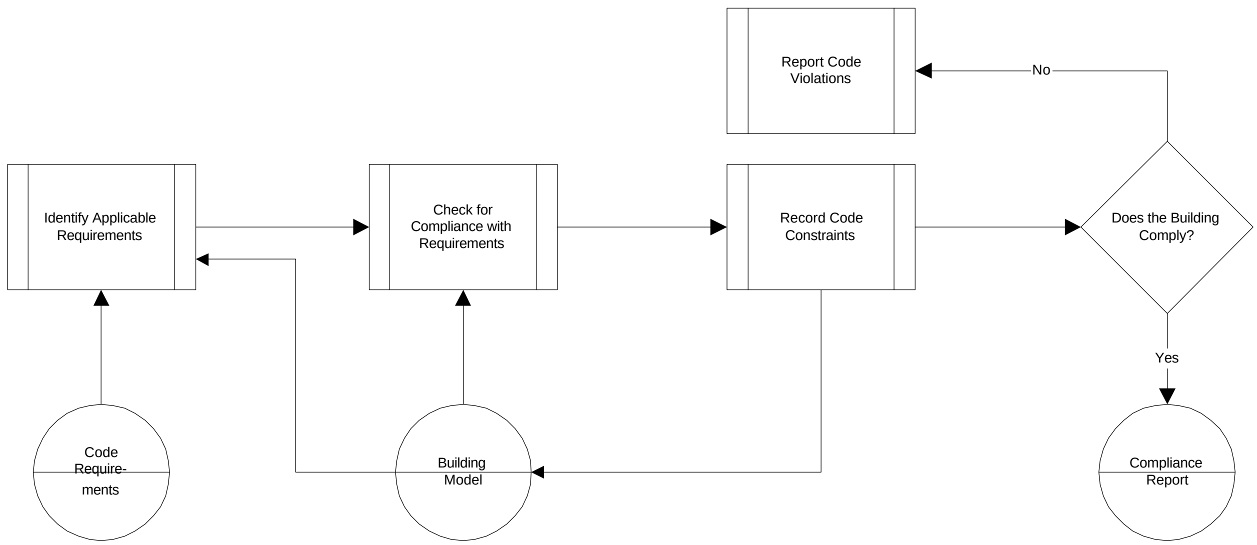
Contributors:

- Rob Briggs, PNNL (NA)
- Tan You Tong, ITI (S)
- Tan Kee Wee, NCB (S)
- Philippe Debras, CSTB (F)
- Robert Amor, BRE (UK)
- Dave Chassin, PNNL (NA)
- Craig Storms, Softdesk (NA)
- Tony Sherfinski, Greenheck Fan (NA)
- Filiz Ozel, ASU (NA)
- David Nack, Nack Engineering (NA)

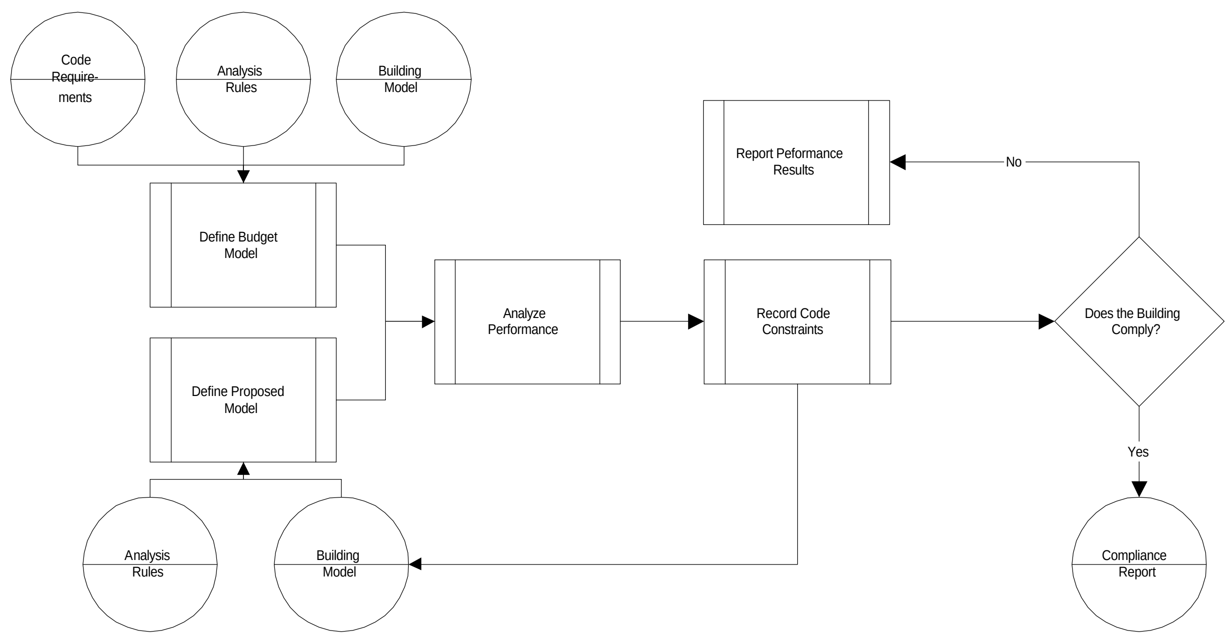
3.10.1.2 Process Diagram

Note: Building codes typically employ two different regulatory approaches: prescriptive requirements and performance requirements. The two-part diagram below illustrates the two different processes corresponding with these two different approaches. Most codes are neither purely prescriptive nor purely performance-based, but rather, contain elements of both types of requirements either in combination or as alternative paths for demonstrating compliance.

Process for Prescriptive Code Requirements



Process for Performance Code Requirements



3.10.1.3 Process Definition

3.10.1.3.1 Overview

Applicable energy codes are normally identified at the programming stage of the project. At the beginning of schematic design, the architect, HVAC engineer, energy consultant, or other designated design team member with responsibility for energy code compliance identifies those code requirements likely to constrain the building design. Depending on the severity of the code constraints, compliance with these requirements may be spot checked as the design process progresses or the energy requirements may be largely ignored until a final compliance check is done, usually at the end of the design development phase of the project.

Most energy code requirements are not strictly prescriptive, but rather constrain the performance of an assembly, subsystem, or major building system. Determining compliance with these requirements frequently requires multiple inputs and some computation. Enabling the necessary data to be managed and manipulated using IFC's will eliminate manual tasks and enable energy code compliance to be checked more easily and frequently during the design process, resulting in compliance at lower cost and with less disruption to the design process. The capability to associate code constraints with objects in the building model will enable design applications to monitor conformance with codes without concurrent operation of code-checking applications. Designers can then focus on the design with confidence that they will be notified if design changes violate code requirements.

3.10.1.3.2 Identify Applicable Code Requirements

Task Description:

This process begins with the intent to demonstrate that a given proposed building design complies with a given energy code. However, specific requirements or even major requirement sections may not apply do to the particular provisions of the code. The determination of which requirements apply is a function of both the code and the project, in particular certain general information about the project such as building location, usage category, and number of stories.

Some specific examples of conditionally applicable requirements are listed below:

- Insulation of the exposed perimeter edges of slab-on-grade construction is not required in climates with fewer than 3,000 heating degree days base 65ºF.
- Basement wall insulation not required when walls are more than one story below grade.
- Either wall or roof insulation requirements may apply to steeply sloping roofs depending on the slope of the roof.
- Certain buildings may be exempted from all envelope insulation requirements based on very low connected loads or the absence of space-conditioning equipment.
- Lighting efficiency requirements do not apply to hotel guest rooms.

Example Usage Scenario: *{{ an example usage scenario for completing this process using data, graphics, etc. from a real AEC project. Try to insure that you have a single project example that spans the full process so that your examples flow from one process task definition to the next. }}*

{{ example scenario description, example scenario description, example scenario description, example scenario description, example scenario description, example scenario description, example scenario description, example scenario description, example scenario description, example scenario description, example scenario description, example scenario description. }}

Sample Graphics

3.10.1.3.3 Check for Compliance with Requirements

Task Description:

This second step for compliance checking with prescriptive code requirements involves a logical comparison of the applicable prescriptive requirements in the code with the corresponding objects and attributes in the building model. This checking process yields both a status result and a code constraint on each of these corresponding building attributes. Commercial energy codes usually contain requirements that pertain to the architectural envelope, lighting systems, and HVAC and service water heating systems. Residential energy codes usually address only building envelope, HVAC, and service water heating.

Example Usage Scenario:

{{ an example usage scenario for completing this process using data, graphics, etc. from a real AEC project. Try to insure that you have a single project example that spans the full process so that your examples flow from one process task definition to the next. }}

{{ example scenario description, example scenario description, example scenario description, example scenario description, example scenario description, example scenario description, example scenario description, example scenario description, example scenario description, example scenario description, example scenario description, example scenario description, example scenario description. }}

Sample Graphics

3.10.1.3.4 Record Code Constraints

Task Description: *{{ thorough description of the task }}*

{{ Description }}

Prescriptive code constraints can be represented as discrete limiting values, which can be associated with a building object and stored in the building model for other applications to utilize. The value of storing code constraint information in the building model (as opposed to simply reporting a compliance result) is that it then can provide persistent guidance to the user and enable user notification when design modifications are made that affect compliance.

In order for this constraint object to be fully useful, it needs carry the following information:

- The object to which the constraint is connected
- The numeric and logical content of the constraint
- Identification of the code to which the constraint belongs
- Identification of the application that established the constraint
- A description of the constraint
- Text to be used in notifying the user about the constraint
- Other objects and attributes on which the value or application of the constraint depends.

Unlike with a prescriptive requirement, performance-based requirement cannot be expressed as discrete constraints on individual object. Rather, the constraint is typically imposed on a system consisting of multiple objects that interact within the code-constrained system. To accommodate performance-based code constraints, it is necessary to attach the constraints to aggregate objects. In addition, many, though not all performance codes employ requirements that are not fixed values but rather are themselves the results of calculations. These requirements tend to have a larger number of dependencies on other objects in the model, and hence require recalculation more frequently than prescriptive requirements.

Example Usage Scenario:

{{ an example usage scenario for completing this process using data, graphics, etc. from a real AEC project. Try to insure that you have a single project example that spans the full process so that your examples flow from one process task definition to the next. }}

{{ example scenario description, example scenario description, example scenario description, example scenario description, example scenario description, example scenario description, example scenario description, example scenario description, example scenario description, example scenario description, example scenario description, example scenario description, example scenario description. }}

Sample Graphics

3.10.1.3.5 Define Budget Model

Task Description:

Compliance checking with performance-based requirements frequently requires that three steps be taken that are not required with prescriptive requirements: defining a budget building model, defining a proposed building model, and analyzing the performance of each. Define the budget model (i.e., the model or building configuration that defines code-minimum performance) is typically performed by implementing prescriptive code requirements into a copy of the description of the proposed design. For example, the code checking procedure may substitute the prescriptive wall and roof insulation requirements for those used in the proposed design. Other assumptions may be imposed to ensure a fair basis for comparison between with the performance results from the proposed model; for example by specifying consistent operating assumptions and energy prices. In addition to implementing these modifications, this task involves translating the representation in the building model to the appropriate representation and format required by the simulation model used to analyze performance.

Example Usage Scenario:

{{ an example usage scenario for completing this process using data, graphics, etc. from a real AEC project. Try to insure that you have a single project example that spans the full process so that your examples flow from one process task definition to the next. }}

{{ example scenario description, example scenario description, example scenario description, example scenario description, example scenario description, example scenario description, example scenario description, example scenario description, example scenario description, example scenario description, example scenario description, example scenario description, example scenario description. }}

Sample Graphics

3.10.1.3.6 Define Proposed Model

Task Description:

A similar process is used to define the proposed model as was used to define the budget model. Most objects in this model are defined directly from the building model entered by the user, however a few assumptions are imposed to ensure a fair comparisons between budget and proposed models. A similar translation is made to required format for the simulation model.

Example Usage Scenario:

{{ an example usage scenario for completing this process using data, graphics, etc. from a real AEC project. Try to insure that you have a single project example that spans the full process so that your examples flow from one process task definition to the next. }}

{{ example scenario description, example scenario description, example scenario description, example scenario description, example scenario description, example scenario description, example scenario description, example scenario description, example scenario description, example scenario description, example scenario description, example scenario description, example scenario description. }}

Sample Graphics

3.10.1.3.7 Analyze Performance

Task Description:

The third step is to calculate the performance of the two models, one representing the design proposed by the user and the other establishing code-minimum performance. While this process description implies a complex process used in the analysis, the basic sequence is applicable to a wide range of analysis procedures. The required calculation may be as simple as a parallel path calculation that combines the rated thermal conductances of two components to determine an overall assembly conductance, or it may be as complex as an annual computer simulation of building energy use for a multi-zone building.

Example Usage Scenario:

{{ an example usage scenario for completing this process using data, graphics, etc. from a real AEC project. Try to insure that you have a single project example that spans the full process so that your examples flow from one process task definition to the next. }}

{{ example scenario description, example scenario description, example scenario description, example scenario description, example scenario description, example scenario description, example scenario description, example scenario description, example scenario description, example scenario description, example scenario description, example scenario description, example scenario description. }}

Sample Graphics

3.10.1.3.8 Compliance Determination and Reporting

Task Description:

Depending on the compliance outcome from the code checking sequence, the user is either notified of the code violation or notified that the design complies and given the opportunity to generate a compliance report for submission as part of the building permit application or inclusion on construction documents. When a prescriptive code requirement is violated, the specific features that violate the requirement are listed for the user along with the corresponding code constraints. Such notification normally prompts the user to modify the design and rerun the compliance check to document compliance. The user may also leave the design unchanged and demonstrate compliance using an alternative, performance-based compliance method.

Unlike with prescriptive requirements, a failure to comply with a performance-based requirement cannot be attributed to the violation of a specific requirement but may depend on a large number of building objects. Results from a performance code evaluation are reported in the form of performance relative to a performance budget. This fact often leads the user to an iterative process to resolve the code violation. The user evaluates various ways of achieving compliance, and often a variety of design and cost issues are considered before a design change is accepted.

Example Usage Scenario:

{{ an example usage scenario for completing this process using data, graphics, etc. from a real AEC project. Try to insure that you have a single project example that spans the full process so that your examples flow from one process task definition to the next. }}

{{ example scenario description, example scenario description, example scenario description, example scenario description, example scenario description, example scenario description, example scenario description, example scenario description, example scenario description, example scenario description, example scenario description, example scenario description, example scenario description. }}

Sample Graphics

3.11 CS-2 Code Compliance Extensions

{{ Process definition and Usage Scenaria for this project not yet available }}

Estimating and Scheduling

3.12 ES-1 Cost Estimating

Processes Defined in this project:

1. **Cost Estimating - Overview:** An overview of the entire cost estimating process.
2. **Object Identification:** Identifying objects in terms of an estimating system.
3. **Task and Resource Modeling:** Modeling the tasks and resources required to build or install an object.
4. **Quantification and Cost Modeling:** Determining quantities and applying cost to the model objects.

3.12.1 Cost Estimating - Overview

3.12.1.1 Introduction

Overview:

This cost estimating usage scenario describes the process of providing cost information to the IFC model from information provided by all objects in the Integrated Model whether physical or logical. Following this 'Cost Estimating - Overview' usage scenario are three other scenarios that examine how the model may be used to implement the processes shown here. Use this scenario to put the following three into an overall cost estimating framework.{{ introduction (in any) to this process definition }}

Process Scope:

- This is an overview of all of the estimating processes. It describes the relationship of its sub-processes; Scope Analysis, Object Identification, Identification of needed Tasks and Resources, Quantification and Costing.

Out-of-Scope:

- The processes described here do not include cost attribute maintenance for actual costs incurred to a project.

Definitions:

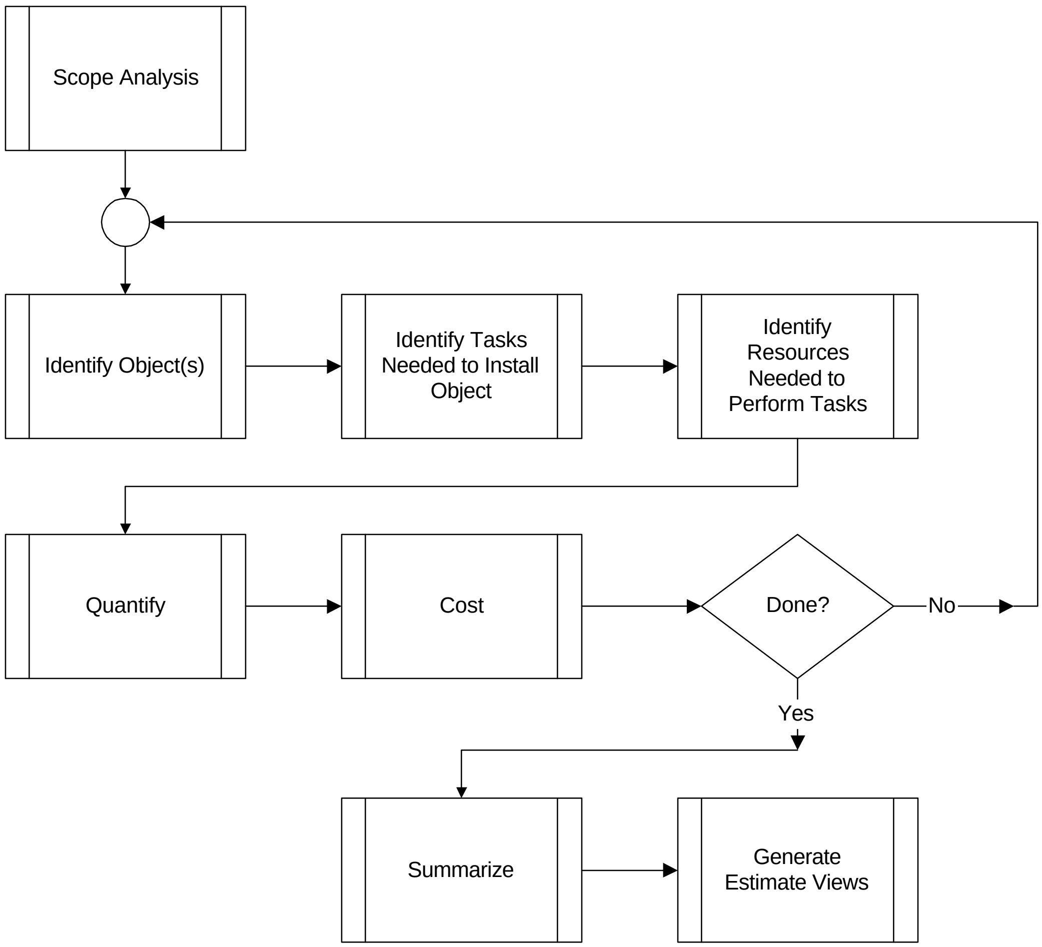
- {{ Term 1 - definition }}

Contributors:

- Mike Cole - NA
- Ray Brungard - NA
- Jeffrey Wix - UK
- Peggy Woodall - NA
- Annette Stumph - NA

- Roger Grant - NA

3.12.1.2 Process Diagram



3.12.1.3 Process Definition

3.12.1.3.1 Overview

Cost Estimating Overview

Scope Analysis

This process looks at the level of detail that will be used in determining the objects that will be used in making the estimate. It provides information pertaining to the overall job, what state the design is in, and possibly for what the estimate is to be used for. This would indicate, for example, whether the estimate was to be detailed or conceptual.

Identify Objects

This process identifies the project objects to be costed and is looped through until all such objects are identified. Such project objects may can be an entire builidng, a section of a building, a space, individual elements (such as door), repeating types of elements (type of door), a process, or a resource. This includes the following sub-processes which are processed in the following order. 1) Select a type of project objects

(Such as doors, windows, walls, etc.), 2) Select a single object instance (One instance of door, window, wall, etc.), and 3) Further identify the object (Determine the ‘type’ of door, wall, ect.).

Identify Tasks Needed to Install Object

This process examines how the object (which was selected and identified in the previous step) is to built in the field and comes up with a set of tasks that are needed to install the object. For instance, a wall may require ‘framing’, ‘sheetrocking’, and ‘finishing’. These tasks can be modeled to come up with more accurate cost estimates. Information about the tasks may be used later in project scheduling.

Identify Resources Needed to Perform Task

Each task will require one or more resources. Resources may include labor (carpenters, electricians, …), equipment (crane, scaffolding, …), and materials (lumber, carpeting, …). Resource objects can be used along with task objects to model the costs of installing an object. Information about the resources may be used later in project scheduling.

Quantify

This process identifies the way in which an object is to be ‘counted’, such as by piece, linear foot, etc. This process is looped through until all objects that have been Identified are Quantified. This includes the following subprocesses which are processed in the following order: 1) Evaluate the Object unit of Measure , 2) Establish the unit of measurement.

Compute Object Quantity

Establish the amount of item to be measured. This includes the counting of discrete objects as well as calculation of quantities from the object’s dimensional information.

Evaluate Resource Units of Measure

Establish the unit of measure for resources needed to install the object.

Compute Resource Quantities

Establish the amounts of each resource that will needed to install the object. This will be calculated using the object’s dimensions and estimated resource usage based on those dimensions.

Cost

This process evaluates the price impact of the object. This process is looped through until all objects Identified are Costed. Using the quantities developed in the previous process, and applying unit costs for the overall object and/or its required resources, the cost impact of the object is calculated.

Summarize

This process takes into account all of the intangibles of an estimate. The following steps need not be taken in any particular order:

Summarize impact to schedule

With regard to an estimate of a change to a project, how does the change, in a general way, impact the schedule. This change may be with respect to other changes, or with respect to the use of the resources involved at the time of the enacted change. This impact to the schedule may have costs not identified with Assess Schedule Impact of any particular object.

Generate Estimate Views

A way to convey the information to the intended customer. This may include a report that organizes and summarize all of the cost information in the model or possibly a browser that would allow a user to look at the cost information for any object in the model.

3.12.2 Object Identification

3.12.2.1 Introduction

Overview:

Identify selected objects in the IFC Project Model and classify them in terms of a cost estimating system. This is done by an estimator at the beginning of the process of costing the objects that will be estimated. This process uses information that was originally entered during architectural design and engineering process to make the classifications. The resulting classifications are then used by a costing system to associate objects in the drawing with objects in its database.

Process Scope:

- Scope Analysis
- Object Identification

Out-of-Scope:

- Task and Resource Modeling
- Quantification and Costing

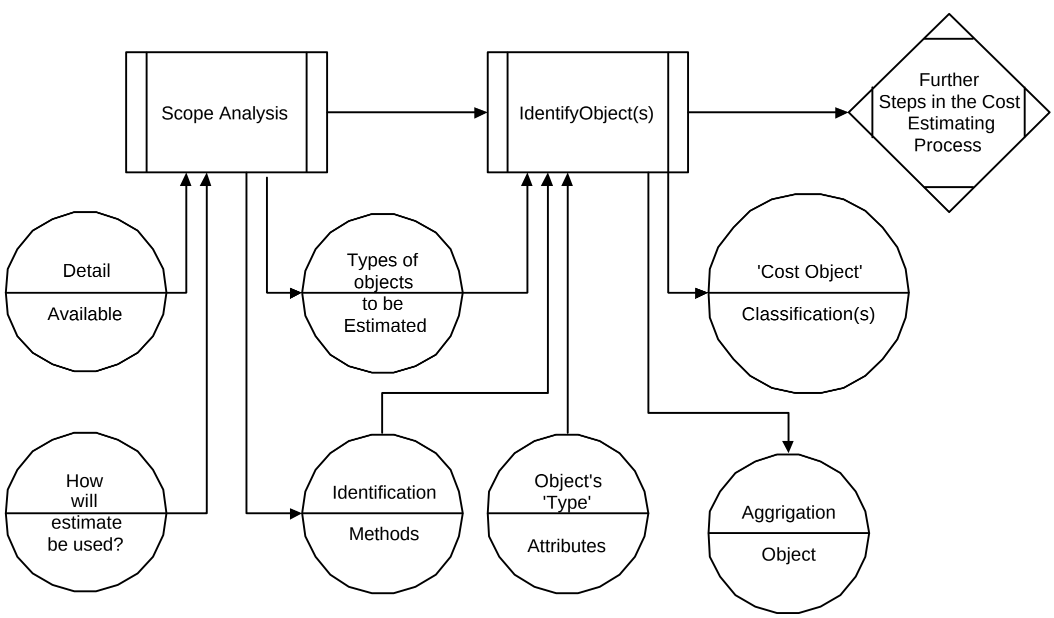
Definitions:

- {{ Term 1 - definition }}

Contributors:

- Mike Cole - NA
- Ray Brungard - NA
- Jeffrey Wix - UK
- Peggy Woodall - NA
- Annette Stumph - NA
- Roger Grant - NA

3.12.2.2 Process Diagram



3.12.2.3 Process Definition

3.12.2.3.1 Overview

Object Identification

There are many cost estimating systems and pricing databases that can be used to determine various aspects of construction cost. They often use classification systems to organize 'cost objects' that model costs of constructing various objects found in an IFC Project Model, such as doors, windows, areas, zones, etc.

To use these estimating systems and pricing databases, the model objects must be mapped to 'cost objects' that can be used to determine their costs. The information found in an IFC Project Model, at various stages of the design process, should allow an application to make these mappings. When a model object is successfully mapped, its 'cost object' classification may be written back to the model object.

3.12.2.3.2 Scope Analysis

Task Description:

This process looks at the level of detail that will be used in determining the objects that will be used in making the estimate. It provides information pertaining to the overall job, what state the design is in, and possibly for what the estimate is to be used for. This would indicate, for example, whether the estimate was to be detailed or conceptual.

Once 'detail available' and 'estimate use' is analyzed, the types of objects to be estimated will be determined. For example, a 'conceptual estimate' may only look at project spaces and zones to come up with a rough initial estimate based on average cost per square meter. At a later stage of the design, more detailed estimating will be possible. The costs of individual walls, doors, windows, etc. can be modeled.

The classifications, attributes, dimensions, and context of model objects may be used to identify them and in turn map them to 'cost objects' in an estimating system. If the identification and classification system is to be highly automated, it must configured at this point to define how information about the objects will be used to map them to a 'cost objects'.

Example Usage Scenario:

At the beginning of the design process, an architect may only know the space and usage requirements of a building. In that case, a *conceptual estimate* would be done based on only this information. If the model contains only space program information, it is determined that only the IfcSpaceProgramme objects will be used in this estimate.

Initial Bubble Diagram

3.12.2.3.3 Identify Object

Task Description:

This process identifies the project objects to be costed and is looped through until all such objects are identified. Such project objects may be an entire building, a section of a building, a space, individual elements (such as door), repeating types of elements (type of door), a process, or a resource. The types of objects to be identified were determined in the 'Scope Analysis'. This includes the following sub-processes which are processed in the order given below:

Select a type of project objects	- Such as doors, windows, walls, etc.
Select a single object instance	- one instance of door, window, wall, etc.
Classify the object	- in terms of the estimating system

Using the object 's type, attributes, dimensions, etc., determine what 'cost object' best models the costing of that object. The most flexible way to perform this task is to use an object browser and classify the objects manually. The quickest way is to define classification rules to automatically classify the objects. Each has advantages and disadvantages. The 'best' way may be some combination of these two.

Once the model object has been mapped to a 'cost object' classification, the classification should be recorded in the model object. This allows the identification process to be separated from the quantification and costing processes.

Example Usage Scenario:

Continuing form the previous example, the object identification process would examine the information in the IfcSpaceProgramme objects. Based on that information, it would select an estimating system object that would best model the cost of space. The process stores the type of the estimating system object in the IfcSpaceProgramme object. Thus, later process do not need classify the object, they need only look up the estimating system classification store on the IfcSpaceProgramme object.

Initial bubble diagram with estimating system classification information.

3.12.3 Estimating Task and Resource Modeling

3.12.3.1 Introduction

Overview:

Model tasks and resources required to install various objects found in the project model. After an object is selected for estimating, its required installation tasks and their required resources are determined. By modeling this information, accurate estimates may be made based on material and labor costs and on predicted production rates. The tasks and resources introduced into the model at this point can be used later by the scheduling process.

Process Scope:

- Identification of tasks required to install an object.
- Identification of resources needed to complete a task.

Out-of-Scope:

- Object identification
- Costing

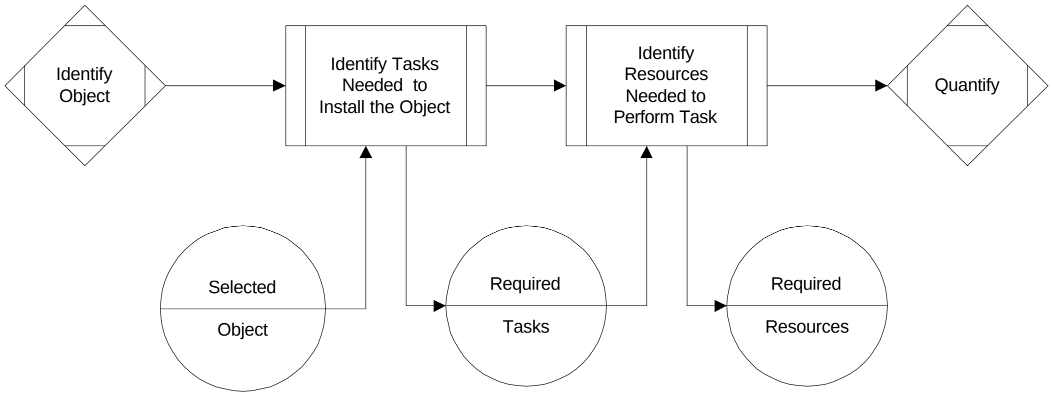
Definitions:

- {{ Term 1 - definition }}

Contributors:

- Mike Cole - NA
- Ray Brungard - NA
- Jeffrey Wix - UK
- Peggy Woodall - NA
- Annette Stumph - NA
- Roger Grant - NA

3.12.3.2 Process Diagram



3.12.3.3 Process Definition

3.12.3.3.1 Overview

Task and Resource Modeling

Once an object has been selected to be estimated, the estimator may want to model the tasks required to install the object. For instance, if a wall has been selected, we may want to model framing the wall, installing drywall, and finishing the wall. The resources needed to perform the tasks can also be modeled. For instance, 'framing the wall' will require carpenters, lumber, nails, etc.

By going to this lower level of detail, estimates that reflect labor and material prices can be produced. The task and resource objects that are added to the model can be used later by applications that schedule the tasks.

3.12.3.3.2 Identify Tasks Needed to Install the Object

Task Description:

Tasks are activities or operations required to place or install any object (permanent or temporary) in the project. To identify the tasks needed, the estimator selects a construction method for the object. The construction method will require one or more tasks to be performed. Task objects will be created and will be referenced the object to be constructed.

Example Usage Scenario:

If we are processing a wall object, several tasks may be identified and added to the model. These tasks may be; framing the wall, putting up sheetrock, finishing the sheetrock, and painting the wall. By modeling each of these tasks, we can later determine the cost of the wall.

Wall diagram

3.12.3.3.3 Identify Resources Needed to Perform a Task

Task Description:

Each task will require one or more resources to be completed. Some resource types include Labor, Material, Subcontractors/Vendors, Equipment, etc. The quantity of the resource that is required and the unit cost of the resource, will contribute to the cost of the task.

The application will either create a resource objects, and/or select ones that already exist in the model. These resources will be referenced by the task to be performed. There may be multiple uses of a resource within the same task.

Example Usage Scenario:

Continuing with the previous example, we now want to determine the resources required to complete each of the tasks. In the case of “framing the wall”, we may need various types of lumber, nails, and carpenters.

Wall diagram

3.12.4 Quantification and Cost Modeling

3.12.4.1 Introduction

Overview:

This is the process of determining quantities of resources required and applying costs to the model objects.

Process Scope:

- Object and resource quantification.
- Costing of resources, tasks and objects.

Out-of-Scope:

- Object identification

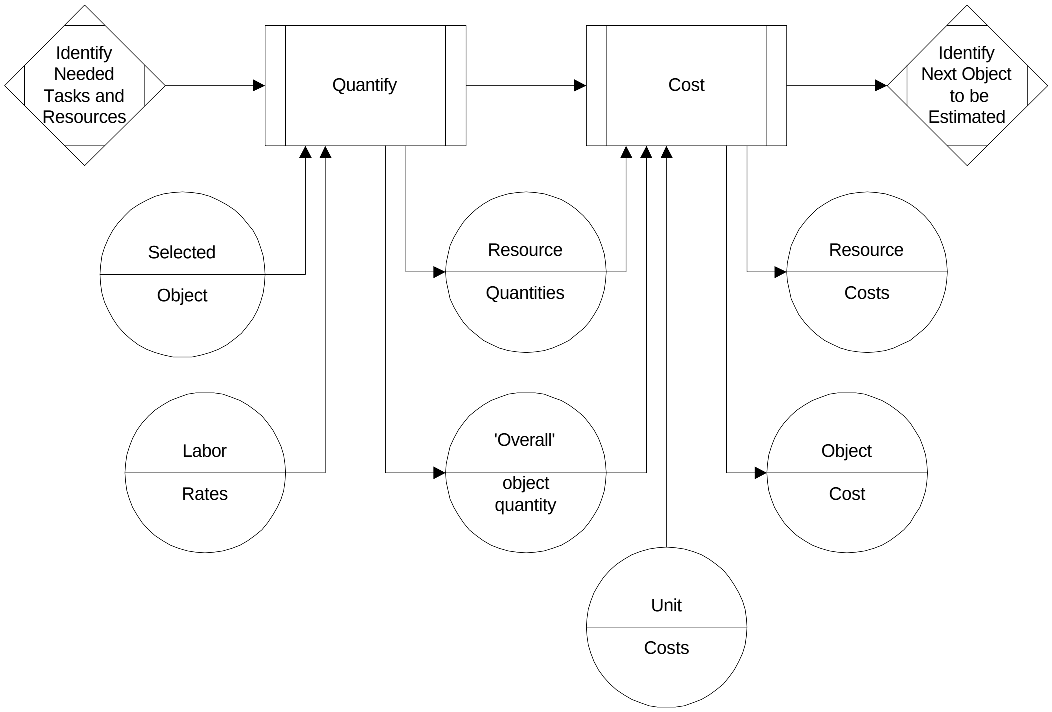
Definitions:

- {{ Term 1 - definition }}

Contributors: *{{ The names and chapters of the domain participants }}*

- Mike Cole - NA
- Ray Brungard - NA
- Jeffrey Wix - UK
- Peggy Woodall - NA
- Annette Stumph - NA
- Roger Grant - NA

3.12.4.2 Process Diagram



3.12.4.3 Process Definition

3.12.4.3.1 Overview

Quantification and Cost Modeling

Once an object has been selected, and the desired tasks and resources have been added to the model, the object should be quantified and costed. The object's dimensions, such as height, length, area, or part count, will be the input for the quantification process.

The object's dimensions may be used to calculate the amount of resources required, or may be used directly to calculate cost based on a unit cost.

Once quantities are determined, unit costs are defined for the objects so that their costs can be calculated. The quantities and costs should be added to the model in a way that is understandable to other processes.

3.12.4.3.2 Quantify

Task Description:

The input to this process is the object that is to be quantified. Depending on the type of object, various dimension attributes will be used to calculate the overall quantity of the object and the quantities of resources required.

The 'overall' quantity of an object is measured in the dimension in which an estimator thinks of it in a 'unit cost' sense. For instance, the overall quantity of a wall might be in linear feet. The overall quantity of a concrete slab might be square feet or cubic yards, depending on how it is being estimated. The overall quantity should be calculated directly from the object's dimension attributes.

The resource quantities are the amounts of the various resources needed to install the object. These quantities are based on the dimension and specification attributes of the object. For instance, a wall's stud count will be based on the length of the wall and the stud spacing and possibly a waste factor stored in the estimating system. The duration quantities for the labor resources will be based on the object's dimension attributes and 'labor productivity rates' stored in the estimating system. Resource quantities should be stored in the task object that 'uses' the resource.

Example Usage Scenario:

To quantify an object, such as a wall, you need to check the model for dimensional information and calculate the length or surface area.

Sample Graphics

To quantify resources, such as materials or labor, you will need the quantity of the wall (length or area) and production rates (usage factors) for the resources. For example, you may need a wall stud for every ½ meter of wall. Or you may need one hour of a carpenters time for every 10 meters of wall.

3.12.4.3.3 Cost

Task Description:

Once the quantities are determined, costs for resource use, tasks, and the primary object can be determined.

Resource costs are calculated based on the resource quantity and the resource unit cost, which originates in the estimating system. The estimating system should update the resource object's unit cost if necessary. It should also update the task's resource use cost.

When all of a task's resources are costed, the resource use costs should be accumulated and the task's overall cost should be updated.

The last cost to be calculated is the primary object's total cost. If no tasks or resources have been attached to the object to model its installation cost, and the object does not have 'parts' which have been costed, the object's cost will be based on its 'overall' dimension, various specifications, and a unit cost originating in the estimating system. The object's 'ProductCost' attribute should be updated with the calculated cost.

If the primary object's cost has been modeled using tasks, resources, or component 'parts' which contain costs, the object's 'ProductCost' should be updated to reflect these factors.

Example Usage Scenario:

If you are doing a conceptual estimate, you may compute cost directly from the quantity of an object. For example, for a certain type of space, you may estimate that the cost is $500 per square meter. Thus from the area of the space and the cost per meter, you can estimate the cost of the total space.

Sample Graphics

If you are doing a more detailed estimate, in which you have done task and resource modeling, you would use the quantities of the various materials and resources and their unit costs to determine their costs. You would then total the costs of the materials and resources to determine the cost of the object.

Facilities Management

3.13 FM-1 Engineering Maintenance

{{ This Project has been delayed for inclusion in Release 3.0 }}

3.14 FM-2 Architectural Maintenance

{{ This Project has been delayed for inclusion in Release 3.0 }}

3.15 FM 3 Property Management (Building Owner's viewpoint)

Processes Defined in this project:

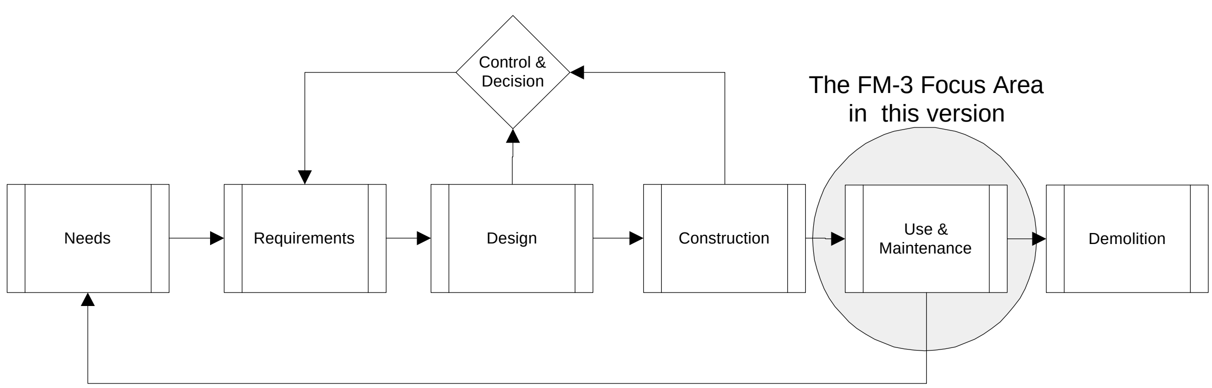
1. Enabling the use of IFC objects in property management
2. Grouping IFC objects
3. Linking the maintenance objects to the IFC objects

3.15.1 Property Management

3.15.1.1 Introduction

Overview:

Property management is a process starting from requirement programming and continuing through the building's life cycle. The tools should facilitate the evaluation and comparison of properties and all costs during the construction and the life cycle. For these purposes the design and product data should be in such a format that it could be combined to the owner's and other external data bases for evaluation and management purposes.



Process Scope:

In this phase the FM-3 project covers just a subset of this process focusing on grouping of spaces and other possible objects for different purposes, like maintenance, administration, public registers, mapping etc. This process is based on objects provided by the design and construction process and uses mainly the attributes in the current model. The main user is the building owner and the benefits is more efficient use of the building data and through this cost savings in the administrative work. This process is starts after the building is completed and is carried out through the whole life cycle of the building.

- Grouping IFC objects
- Linking the maintenance objects to the IFC objects

Out-of-Scope:

- Instructions for the maintenance
- Evaluation methods

Definitions:

- Group: a set of selected objects and / or groups
- Maintenance object: an object containing description, classification and maintenance history of linked IFC building elements

References:

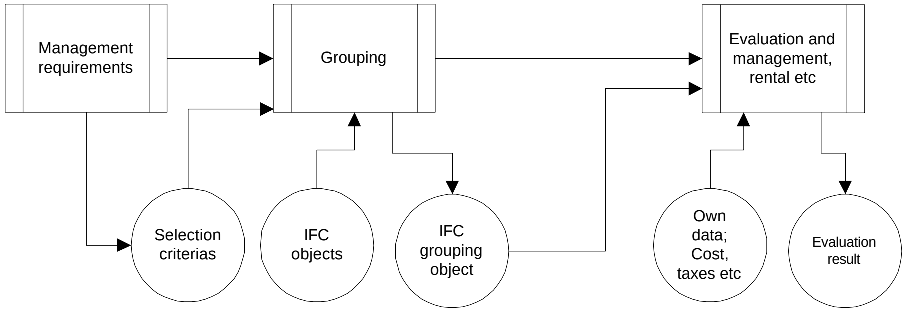
- Version 1.0 IFC Model
- CB-1 Client Briefing
- ES-1 Cost Estimating
- (FM-1 and FM-2 Maintenance, in version 3.0)
- FM-4 Occupancy Planning

Contributors:

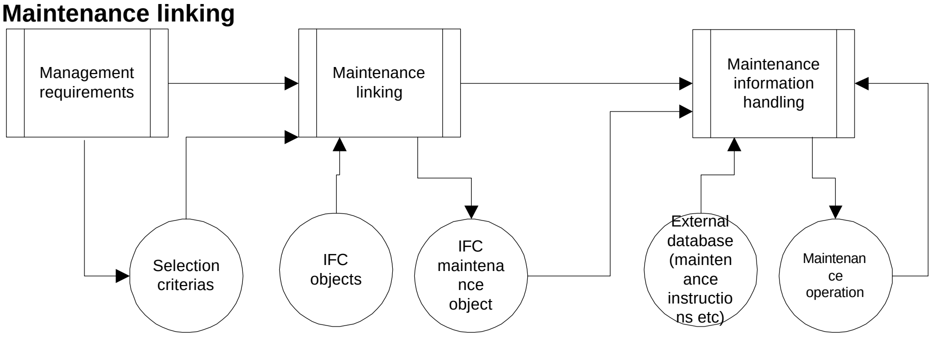
- Poul Sorgenfri Ottosen (AAU, Danish Department of Education) (Nordic)
- Jan Karlshoej (Carl Bro A/S) (Nordic)
- Arto Kiviniemi (VTT Building Technology) (Nordic)
- ?

3.15.1.2 Process Diagram

Object grouping



Maintenance linking



3.15.1.3 Process Definition

3.15.1.3.1 Overview

The need for grouping can be caused by any management purpose, like a new department, workgroup, cleaning area, renovation, fire zone etc. In this process the property manager can create new groups from selected objects. These groups can be used for any administrative or management purposes. All material or quantitative information is calculated from the IFC model. The model information can be used together with the owner's own or other external database information to evaluate operational costs or other needed values.

3.15.1.3.2 Grouping IFC objects

Task Description:

The first task is to define the grouping purpose, which defines the classification of this group. Then the objects for new groups can be selected through various methods:

- any objects selected by the user
- filtered objects (type, properties or other selection key) selected by the user
- filtered objects in the whole model

After the selection is completed the user can give a description to the group.
If the selected objects already belong to some group with the same classification, the application should warn the user about it and ask for instructions for further operations.

When the groups are formed the user can use those as the selection criteria for different operations and reports. All IFC object data should be available through these selections.

Example Usage Scenario:

The cleaning areas in the building needs to be defined. Different materials need different operations. The materials on surfaces can be recognized from the IFC model and picked automatically. After this the selected objects can be divided to proper sizes for operations and grouped to one unit. The classification and description of the unit enables easy administration, visualization and reporting of these units.

3.15.1.3.3 Linking the maintenance objects to the IFC objects

Task Description:

First task is to define the selection criteria for a maintenance group. Then the objects for a new group can be selected through various methods:

- any objects selected by the user
- filtered objects (type, properties or other selection key) selected by the user
- filtered objects in the whole model

After the selection is completed the user can give a description to the new maintenance group.
If the selected objects already belong to some maintenance group, the application should warn the user about it and ask for instructions for further operations.
When the maintenance groups are formed the user can use those as the selection criteria for different maintenance operations and reports. All maintenance data is stored in the maintenance object and the IFC object data should be available from the actual objects.

Example Usage Scenario:

The different window types in the building are linked to a maintenance object. The guarantees, maintenance periods and maintenance history of these windows is stored in the maintenance object. The property manager can check from this information when maintenance operations should be done and if all necessary operations are made according to the guarantee terms. The grouping mechanism is identical to the grouping activities.

3.16 FM-4 Facilities Management

3.17 FM4 Occupancy Planning *(incl. Design and layout of workstations)*

Processes Defined in this project:

1. Occupancy Planning
2. Design of Workstations
3. Floor Layout of Workstations for an Open Office

3.17.1 Occupancy Planning

3.17.1.1 Introduction

Overview:

The occupancy planner (includes interior designers, facilities managers, architects, furniture dealers' designers, etc.) applies standards during the assignment of people and organizations to interior spaces. It also involves the planning and moving of building assets such as equipment and furniture. This process occurs during the initial planning of space occupancy, and whenever that occupancy needs to change (company reorganization, company growth, or new hires, etc.). The layout and design of typical workstations can be sub-processes of the occupancy planning when it involves systems furniture planning for open offices. These processes require information about the building floor spaces. They also generate space occupancy data for future use of office planning.

Automatic input and utilization of the IFC supported object data, such as building elements and spaces as well as FF&E and occupants, may improve the efficiency of the processes. New objects generated will also be IFC compliant so that they can be used by varies FM processes during the operation of the facility.

Process Scope:

- Evaluation of spaces
- Move of people and FF&E

Out-of-Scope:

- Design of workstations
- Floor layout of workstations
- Stacking and blocking
- Work and purchase order tracking process

Definitions:

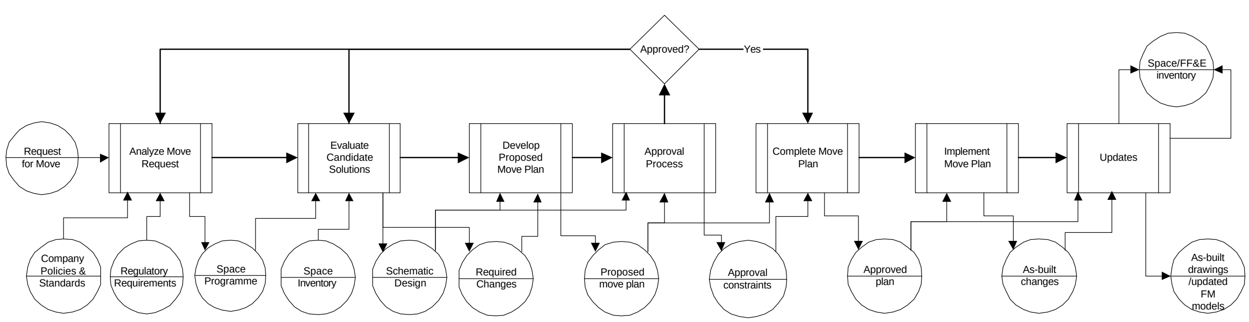
- FF&E: furniture, fixture, and equipment that is movable.
- Schematic Design: the conceptual allocation of space to define adjacencies, required functions defined by area and circulation paths.
- Move Plan: a plan that is used in Facilities Management for occupancy planning, moving people and FF&E around.

References:

Contributors:

- Sandy Anderson (IBM) (NA)
- Rick Bartling (Herman Miller) (NA)
- Vicky Borchers (MKS) (NA)
- Francois Grobler (USA-CERL) (NA)
- Jeff Laba (IBM) (NA)
- Paul Lewis (Visio) (NA)
- Elizabeth Menard (Naoki Systems) (NA)
- Richard See (Autodesk) (NA)
- Kevin Yu (Naoki Systems) (NA)

3.17.1.2 Process Diagram



3.17.1.3 Process Definition

3.17.1.3.1 Overview

The occupancy planner (includes interior designers, facilities managers, architects, furniture dealers, etc.) applies standards during the assignment of people and organizations to interior spaces. This process occurs during the initial planning of space occupancy, and whenever that occupancy needs to change (company reorganization, company growth, etc.).

3.17.1.3.2 Analyze Move Request

Task Definition:

Evaluate request with respect to occupant information, company policies, regulatory requirements. Identify FF&E required for the occupant, and generate space programme.

Example Usage Scenario:

The first step is to analyze the move request. In this step, the occupancy planner evaluates the request with respect to occupant information, company policies, and regulatory requirements. This step may identify the FF&E required, and finally generate the space program for the request.

3.17.1.3.3 Evaluate Candidate Solutions

Task Definition:

Compare space programme to available spaces to find candidate solutions including the changes of spaces and FF&E.

Example Usage Scenario:

The second step is to evaluate candidate solutions. The space program from the last step is used to block plan available spaces, and find candidate solutions that include the changes of spaces and FF&E. This process will also result in schematic designs.

3.17.1.3.4 Develop Proposed Move Plan

Task Definition:

During the design and generation of drawings, we allow for client review and approval. Define temporary staging areas, generate schedules, identify sources of all FF&E required and generate a cost estimate.

Example Usage Scenario:

An occupancy move plan should be developed in this step to allow for client review and approval. A list of all FF&E required is created. A preliminary work schedule and a cost estimate will be included in the plan. The schematic design used in the last step will also be included in the plan package.

3.17.1.3.5 Approval Process

Task Definition:

Occupant and management review proposed move plan and either approve (possibly with constraints) or rejects --> revert to previous steps.

Example Usage Scenario:

The approval process involves the review of proposed plan. This process could either approve (possibly with constraints) or rejects. In the case of rejection, it is possible that the move request is re-analyzed or the candidate solutions are re-evaluated.

3.17.1.3.6 Complete Move Plan

Task Definition:

Modify proposed plan to comply with constraints. Generate work orders and purchase orders.

Example Usage Scenario:

If the plan has been approved, there is a need to modify the proposal as with the constraints suggested. The work orders and purchase orders will be generated, and a new plan will be developed. Based on the new plan, bills-of-materials for the purchase of new FF&E will be produced.

3.17.1.3.7 Implement Move Plan

Task Definition:

Purchase FF&E. Perform work orders. Deal with change orders. Move the occupant.

Example Usage Scenario:

The space occupants including the existing FF&E will be moved. During the implementation, as-built changes will be summaried and possibly updated into the original move plan. The implementation will eventually result in new or revised space and FF&E inventories.

3.17.1.3.8 Updates

Task Definition:

Revised documentation and databases to reflect new and revised spaces and assets.

Example Usage Scenario:

Finally, documents and databases of space and asset (i.e. FF&E) inventory will be updated to reflect the changes.

3.17.2 Design of Workstations

3.17.2.1 Introduction

Overview:

The facility manager (also interior designers, architects, furniture manufactures and designers, etc.) designs typical workstations to be used by office staff. The workstations designed could be used as company standards and be selected in the layout of the systems furniture. This process could also occur in the entire process of occupancy planning in an organization.

Process Scope:

- Systems furniture design
- Approval of design

Out-of-Scope:

- Stacking and blocking
- Standardizing workstations
- Layout of workstations
- Design of workstation groups

Definitions:

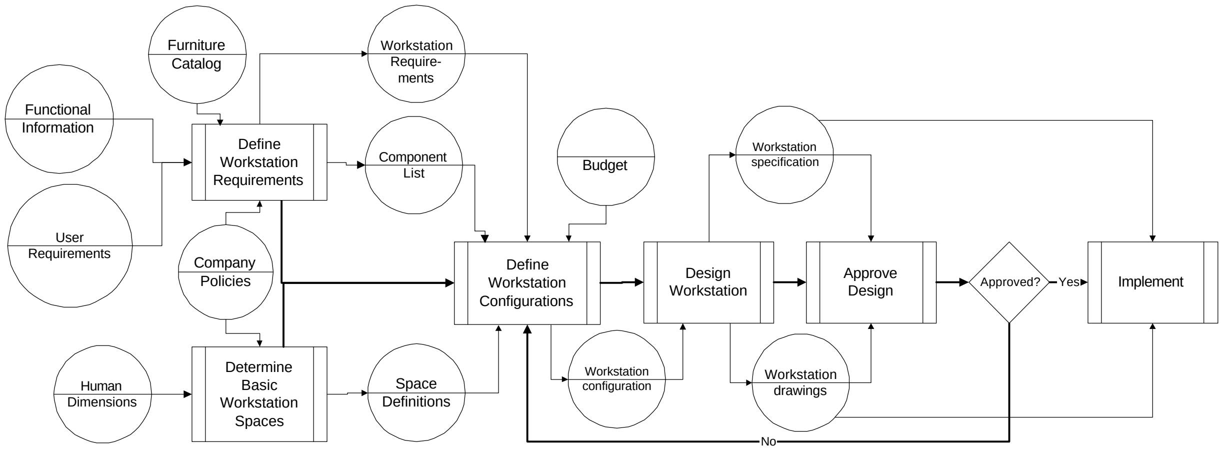
- FF&E: furniture, fixture, and equipment that is movable.
- Workstation: a bound space assembled by systems furniture with necessary office equipment to house one or a few people to perform tasks.
- Systems furniture: is meant to represent furniture systems that integrate both modular and free-standing furniture with independent panels hanging furniture components such as work surfaces, storage, and so on.

References:

Contributors:

- Sandy Anderson (IBM) (NA)
- Rick Bartling (Herman Miller) (NA)
- Vicky Borchers (MKS) (NA)
- Francois Grobler (USA-CERL) (NA)
- Kevin Yu (Naoki Systems) (NA)

3.17.2.2 Process Diagram



3.17.2.3 Process Definition

3.17.2.3.1 Overview

The facility manager (also interior designers, architects, furniture manufactures and designers, etc.) designs typical workstations to be used by office staff. The process starts from defining the functional requirements of the workstation based on the work types of the employees who use the workstation. The workstation to be designed must also meet the requirements of basic human dimensions for spaces. Special requirements such as a wheelchair must be considered. The design drawings and specifications should be produced based on the configurations of the workstation components and equipment. Final design must be approved before implementation.

3.17.2.3.2 Define Workstation Requirements

Task Description:

Define the basic component and equipment types, the security, privacy and special requirements according to the employee type, work types, and company policies, etc..

Example Usage Scenario:

The first step is to define the functional requirements of the workstation. This requires information about type of the user (e.g. a computer programmer), and the type of work (e.g. design and programming) he or she performs. A functional information worksheet (see Table 1) can be used for collecting the information. Ergonomic requirements for particular types of users will also be considered such as that a wheelchair must be used, or the height of an individual. Some companies may also want to apply some special company policies for this process; examples are style of furniture for managers, etc.. Based on the information provided, a list of basic furniture components will be generated such as types of work surfaces, file storage, panel partitions, lighting and seating. In addition, a list of office equipment types will also be created. For example, a programmer will need a computer; and based on the work types, the computer may need a modem. The workstation requirements will be summarized that include security requirements (e.g. files must be locked), electrical and telecommunication requirements (e.g. 3-circuit, dedicated, network type, etc.), privacy requirements (e.g. visual privacy), and any types of special requirements such as aesthetic requirements. Table 2 shows a sample list of workstation component and equipment types for a computer programmer.

Table 1: Sample Functional Information Worksheet

Employee Name:	Jack Smith		
Employee Type:	Computer Programmer		
Work Task Description:	Analysis	Programmin g	interne t access
Storage Items:	Books References Accessories	References	
Components Required:	computer surface write/read surface storages	computer surface	mode m
Equipment Required:	PC computer	PC computer	
Average Weekly Hours:	10	20	2.5
Special requirements:	no		

Table 2: List of Component and Equipment Types

Furniture Component:	
worksurface	writing & reading PC computer file references accessories
Storage	filing storage reference storage
Seating	main chair (1)
Panel type	moderate privacy enclosure
Lighting	ceiling lighting
Equipment:	
PC Computer	1 (with modem)
Special Requirements:	
no	

3.17.2.3.3 Determine Basic Workstation Spaces

Task Definition:

Define spaces of the workstation (including circulation space inside of the workstation) according to the basic requirement of human dimension standards, and company policies.

Example Usage Scenario:

The basic spaces according to the human dimension standards requirements will be determined in this step. This step can be performed in parallel with the first one. Table 3 shows some examples of human dimension requirements for a basic workstation. One may also want to apply some company policies to this step.

Table 3: Human Dimension for Basic Workstation

Zone	Dimension (inch)
Worktask Zone	66 - 70
Chair Clearance Zone	66 - 70
Circlulation Zone	24
Worksurface Height	30 - 36

3.17.2.3.4 Define Workstation Configurations

Task Definition:

Finalize all workstation components with all detailed dimensions and material information, and spaces.

Example Usage Scenario:

After the above two steps have been finished, detailed workstation configurations will be designed, which include all the information about the components (see Table 4), equipment, and spaces (i.e. their dimensions, materials, space footage, and even brands, suppliers, models, colors, etc.).

Table 4: Sample Component Configuration of Workstation

Item Description	Dimension			Hunging Height	Finish/ Color		
	Height	Width	Depth			Quantity	Remark
Overhead storage with task light	1.5ft	4ft	1ft	6.3ft	walnut	2	
Worksurface rectangle		4ft	2ft	2.5ft	walnut	2	
Worksurface square corner		4ft	4ft	2.5ft	walnut	1	
Chair		2ft	2ft		blue fabric	1	
Panel	6.5ft	4ft			grey fabric	4	

3.17.2.3.5 Design Workstation

Task Description:

Produce the workstation drawings and define the specifications according to the configurations.

Example Usage Scenario:

Based on the configurations defined in the last step, design drawings and specifications will be produced in this step. See Figure 1 for an example of a workstation drawing.

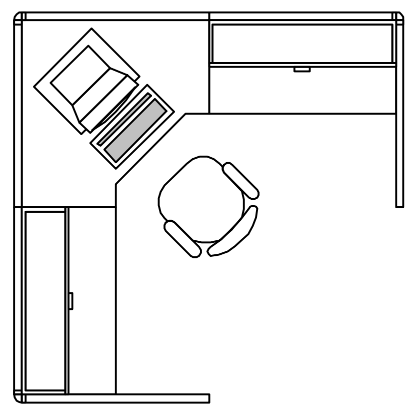


Figure 1: Sample Workstation

3.17.2.3.6 Approve Design

Task Description:

Approve Design.

Example Usage Scenario:

This process examines the design produced in the last step and attempts to approve it.

3.17.2.3.7 Implement

Task Description:

Implement the design.

Example Usage Scenario:

After the design has been approved, the implementation will be executed.

3.17.3 Floor Layout of Workstations for an Open Office

3.17.3.1 Introduction

Overview:

The facility manager (also interior designers, architects, or furniture dealers, etc.) designs the layout of the workstations for an open office. The purpose of the design is to organize the individual workstations into workstation groups with each usually representing a departmental unit and performing a certain type of function as a whole such as marketing. The workstation groups are assembled workstations connected by shared vertical panels. In order to group the workstations, employee work interaction patterns must be captured and block plan mechanism may be used. The process is part of the entire floor furniture and equipment planning for the department(s), and occurs after typical individual workstations have been designed.

Process Scope:

- Floor blocking

Out-of-Scope:

- Bubble diagram design
- Design of workstations
- Standardizing of workstations
- Stacking
- Approval process of the design

Definitions:

- FF&E: furniture, fixture, and equipment that is movable.
- Workstation: is a bound space assembled by systems furniture with necessary office equipment to house one or a few people to perform tasks.
- Systems furniture: is meant to represent furniture systems that integrate both modular and free-standing furniture with independent panels hanging furniture components such as work surfaces, storage, and etc..
- Workstation group: physically adjacent workstations that together perform a certain function, such as marketing, or computer programming.
- Floor blocking: the process of designing block plans that are two-dimension horizontal layout in a floor plan. Each block is a large bound zone that contains one or more work groups that represent a functional unit such as a department, etc..
- Bubble Diagram: a graphical diagram that depicts the adjacency relationships of work groups (e.g. departments) using circles and lines without floor space restrictions. The circles (i.e. bubbles) are used to represent the work groups by their size, color, or shading meaning the size and name of the group. The lines are used to represent the closeness of the adjacency usually by their thickness.

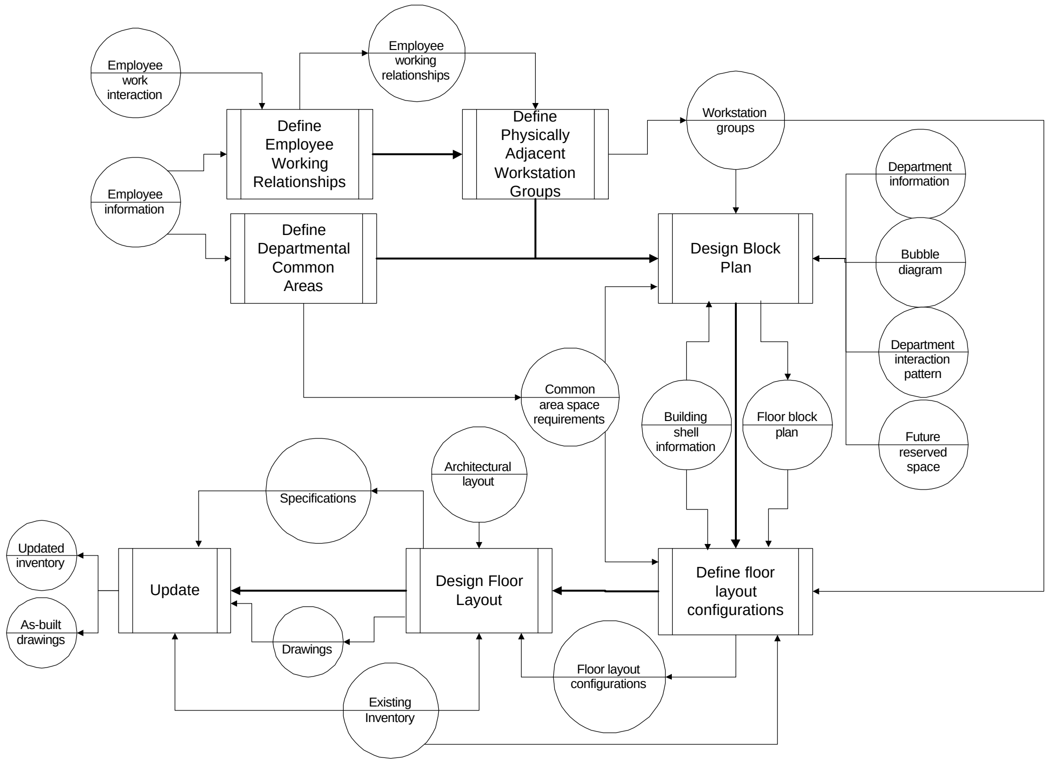
References:

- Julie K. Rayfield, “The Office Interior Design Guide – An Introduction For Facilities Management & Designers”

Contributors:

- Sandy Anderson (IBM) (NA)
- Rick Bartling (Herman Miller) (NA)
- Vicky Borchers (MKS) (NA)
- Francois Grobler (USA-CERL) (NA)
- Kevin Yu (Naoki Systems) (NA)

3.17.3.2 Process Diagram



3.17.3.3 Process Definition

3.17.3.3.1 Overview

The facility manager (also interior designers, architects, or furniture dealers, etc.) designs the layout of the workstations for an open office. The process starts from defining the employee working relationships so that closely related workstations can be adjacently assembled into workstation groups. Common departmental areas such as circulation or service areas must be considered. The adjacency relationships between the departments or workstation groups must be determined. It is usually necessary and efficient to use block plan mechanism to mark the floor area into different and big plane blocks with each representing a departmental unit, such as a research department. Workstations and groups will then be fit into certain blocks. Actual design drawings and specifications of the workstation layout will be produced based on the workstation layout configurations. The design must be approved before implementation.

3.17.3.3.2 Define Employee Working Relationships

Task Definition:

Define the individual employees working interaction patterns and meeting frequencies according to the work they perform.

Example Usage Scenario:

An employee working interaction pattern summary is produced in this step. This summary includes information such as department name of the employees, with whom one has interaction with the other, how many times of such interaction, and average duration of each meeting. A worksheet (see Table 5) can be used for collecting information and interaction analysis.

Table 5: Sample Worksheet of Employee Interaction

Employee Information			Interaction	With whom			Average	Daily
Name	Dept.	Position	Description	Name	Dept.	Where	Dur.	Freq.
Jack	Dev.	System Designer	Program Corporation	Tony	Development	either office	30 min.	0.5
			Consulting	Kevin	Research	Kevin	30 min.	0.25
			Customer Requirement	Linda	Marketing	either	5 min.	1
			Approval	Jeff	Project Manager	Jeff	20 min.	0.25
…		…	…	…	…	…	…	…

3.17.3.3.3 Define Physically Adjacent Workstation Groups

Task Definition:

Define the functional workstation groups according to the individual employees working relationships. A group consists of one or a few different types of typical workstations that have close working relationships, frequent interactions, and perform the same kind of function.

Example Usage Scenario:

Once the employee working interactions have been determined, the physically adjacent workstation groups can be defined with each providing a certain working function (e.g. development group). Each workstation group consists of one or a few different types of typical workstations that have close working relationships, frequent interactions, and perform the same kind of function as a whole. Adjacent workstation groups are typically connected by shared vertical panels and necessary connecting accessories. See Figure 2 for an example.

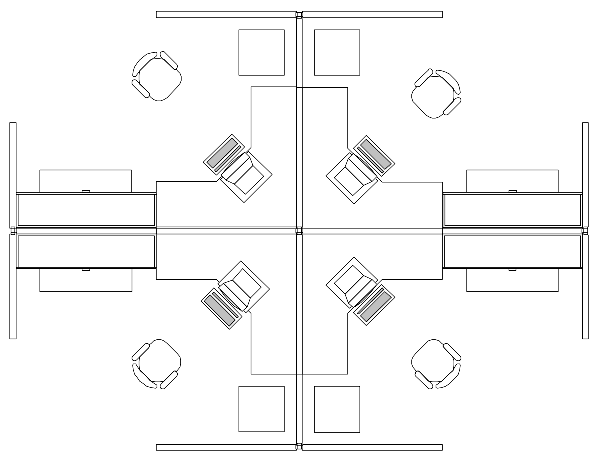


Figure 2: Sample Workstation Group

3.17.3.3.4 Define Departmental Common Areas

Task Definition:

Define the areas that are shared by all employees in the department, such as common circulation and conference rooms, etc.

Example Usage Scenario:

The departmental common area such as the common circulation areas, conference rooms, printing service center, etc.is defined. This step can be performed independently with the first one. The common area requirements will be used for floor layout design in later steps.

3.17.3.3.5 Design Block Plan

Task Definition:

Segment large spaces for workstation groups according to the relationships between the workstation groups, and the relationships between departments in case of multiple departments. Floor geometry constraints such as column grids, ceiling grids, window grids, the space footage must be taken into consideration. A floor block can contain one or more workstation groups, or one or more workstations.

Table 6: Sample Adjacency Matrix

Accounting	3	2	2	1	1
	Marketing	3	2	2	1
		Executive	3	2	2
Adjacency Rating Code:			**Operation**	2	2
1. Undesirable				**Research**	3
2. Desirable					**Development**
3. Essential					

Example Usage Scenario:

After the above three steps, a floor block plan can be designed according to the relationships between the workstation groups, and the relationships between departments in case of multiple departments. The relationships between the workstations can be determined through the 'Adjacency Matrix' mechanism (see Table 6). Different adjacency rating scheme can be used. Building shell information such as column grids, ceiling grids, window grids, the space footage should be essential as input information for this design. Eventually, each block could contain one or more workstation groups as well as individual workstations that do not belong to any defined group. Figure 3 is a sample that shows a portion of a floor block diagram.

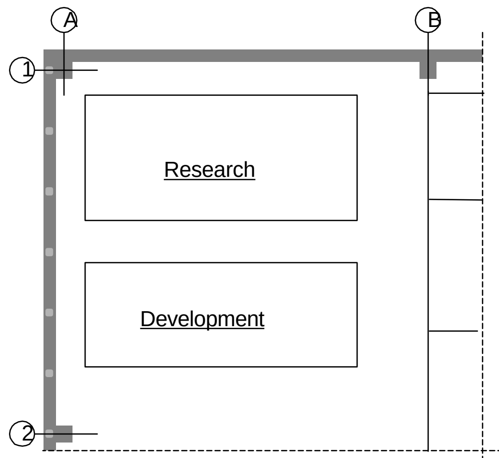


Figure 3: Sample Floor Block Diagram

3.17.3.3.6 Define Floor Layout Configurations

Task Definition:

Define all the detailed footage of all the workstations, workstation groups and departmental boundaries.

Example Usage Scenario:

In order to do the layout design, detailed floor layout configurations must be defined, which includes all the detailed footage of all the workstations, workstation groups, and departmental boundaries on the floor. The furniture system chosen during the floor layout configuration will affect the workstation boundaries.

3.17.3.3.7 Design Floor Layout

Task Definition:

Produce the workstation layout drawings and define the specifications.

Example Usage Scenario:

Based on the configuration, the final step is to perform the design of floor workstation layout. From this step, drawings

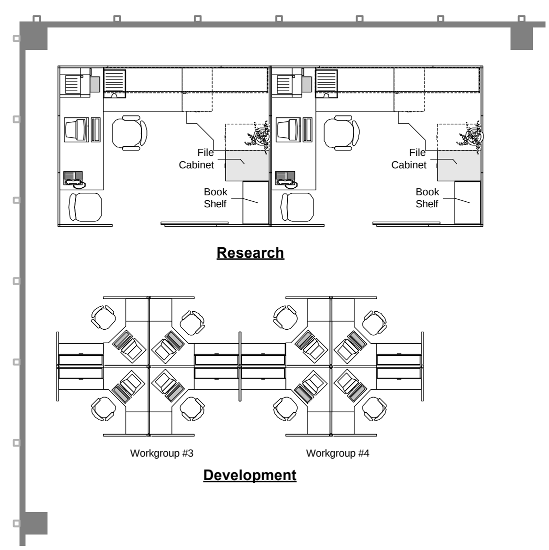


Figure 4: Sample Floor Layout of Workstations

and specifications will be produced. Once approved, the design is implemented. Figure 4 shows a portion of a floor layout of workstations as an example.

3.17.3.3.8 Update

Task Definition:

This is an on-going process that occurs during the course of design implementation. Inventories are updated. Drawings are changed and as-built drawings are produced overtime.

Example Usage Scenario:

As-built drawings are updated; any existing inventories are updated with left over furniture from the job.

Simulation

3.18 SI-1 Visualization

Project process list:

- Architectural Visualization.

3.18.1 Architectural Visualization

3.18.1.1 Introduction

Overview: In the design of a building or other structure, the architect or designer may want to see what the building or the structure will look like, or may want to render images for the client's benefit. Such visualization may be desired at any time from the earliest architectural design or retrofitting to the final interior design. Visualization is the key to solving lighting and daylighting design problems, and is also important in assessing building performance and human comfort issues.

Process Scope:

- Selection of surface materials
- Selection of lighting
- Rendering

Out-of-Scope:

- Process of acquisition of space/building geometry
- Photometric information that may be generated by the application used in the simulation

Definitions: No special industry-specific definitions or acronyms are used in this section

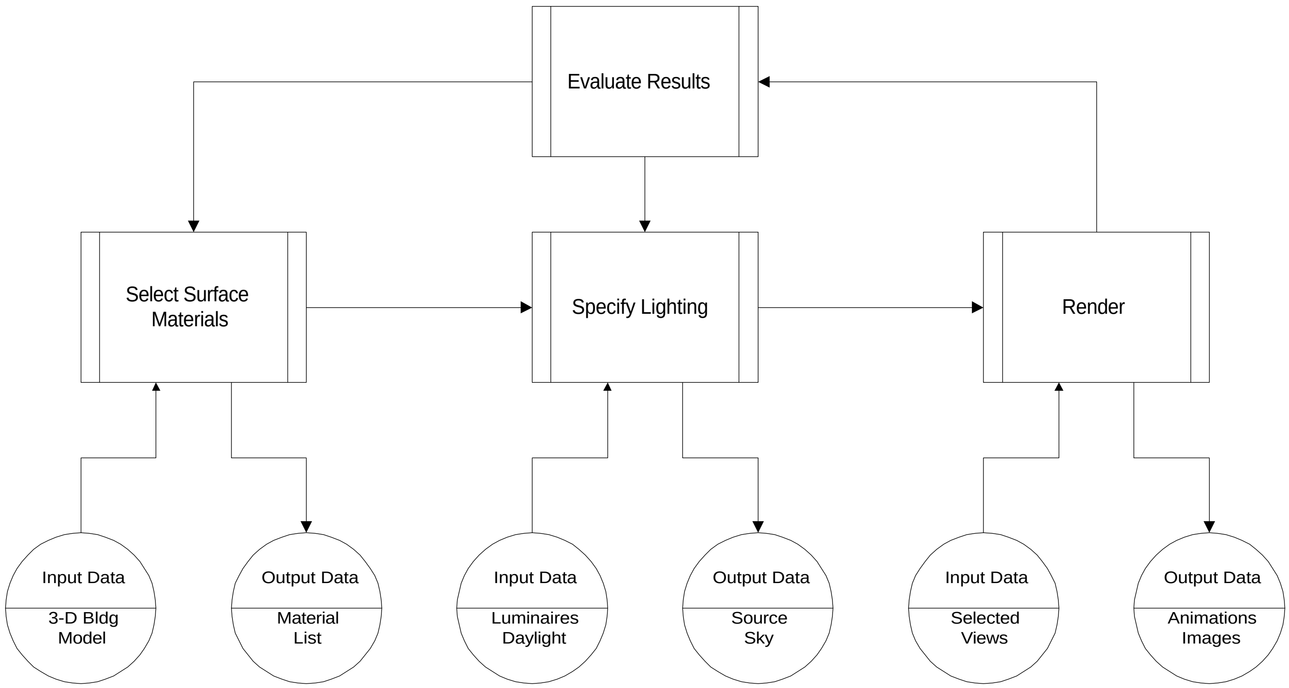
References: N/A

Contributors:

- Vladimir Bazjanac (LBNL) (NA)
- Greg Ward (LBNL) (NA)

Note: Contributors listed here are people who have contributed to the definition of this process to-date; more names may be added later.

3.18.1.2 Process Diagram



3.18.1.3 Process Definition

Overview

The input information for the selection of materials consist of the three-dimensional representation of space or building geometry. Each of the surfaces that affect the rendering is associated with a particular material, for which reflectance, transmittance, color, pattern and texture are defined.

To perform a visual simulation, the user selects and places light sources (luminaires) in three-dimensional space, and specifies daylight conditions. Light source configuration and light distribution data are selected from manufacturers' catalogs. The sun and sky conditions (sky distribution and solar position specific to time in the simulation) are taken from a set of quantitative models (including daylight models) appropriate to the building site.

To define the rendering, the user also specifies a point in three-dimensional space from which the space or the building are viewed. The user may also specify the animation path, should he wish to create an animation. The output from the simulation are two-dimensional (floating point) color images, luminance and isolux contour plots, and/or animations.

The input of three-dimensional geometry description of the space or the building, if done manually, is very time-consuming an error prone. So is, to a lesser extent, the manual input of material and surface properties. If these data are originally input into IFC-compliant CAD software and data bases, the automatic acquisition of the data will reduce input preparation time by orders of magnitude and virtually eliminate input data error. This will substantially reduce the cost of use of visualization tools and make the daily application of such tools in building design and construction attainable.

Task Description

The user loads the space or building description (in form of 3-D building geometry), selects the materials for each surface that affects the rendering, defines the source(s) of light and the associated attributes, selects a view-point, defines the parameters of rendering and executes the simulation.

Example Usage Scenario

The architect has redesigned the space to serve as a computer classroom and has, together with the interior designer, planned the layout of computers and monitors in the space. Since three of the four walls that define this space have large windows, reflection and glare from monitor screens may render this layout unusable. To find the extent of possible reflection and glare, the architect uses a high-end visualization tool to generate a photo-accurate image of the space and all furnishings. The resulting image clearly conveys that reflection from computer monitors will be unacceptable in the current layout. The architect and the interior designer will have to change the orientation (position) of monitors, introduce effective blinds or drapes, or change the layout of the space.

Structural Engineering

3.19 ST-1 Steel Frame Structures

{{ Process definition and Usage Scenaria for this project not yet available }}

3.20 ST-2 Concrete Frame Structures

{{ This Project has been delayed for inclusion in Release 3.0 }}

3.21 ST-3 Sub-Structure Design

{{ This Project has been delayed for inclusion in Release 3.0 }}

3.22 ST-4 Structural Loads Definition

{{ Process definition and Usage Scenaria for this project not yet available }}

Cross Domain Projects

3.23 XM-1 Referencing External Libraries

{{ This Project has been delayed for inclusion in Release 3.0 }}

3.24 XM-2 Project Document Management

Domain process list: A list of processes being enabled by this domain.

3.24.1 Project Document Management

3.24.1.1 Introduction

Overview: Project Document Management refers to all information pertaining to the documents used to estimate, bid, purchase, and manage the building process as well as for use within the Facilities Management domain. This data identifies the document, the author of the document, changes to the document since the last change, and relationships to other documents.

It has been suggested to the group that the first concentration of our should be on the Contract Drawings represented in the model. It is acknowledged that this is only a small subset of the related documents of the model.

Process Scope: An itemization of the process tasks that are within scope for this process for this version:

Create Drawing View:

Retrieve Drawing View:

Out-of-Scope: An itemization of any process tasks that have been purposely omitted from this process

All NonCAD Document Views (such as Specifications, Change Orders, etc.)

Definitions: Any industry-specific definitions or acronyms that are used in this section

Bulletin: a collection of Drawings, Specifications, Sketches, and instructions transmitted to the Project Team from the Architect in order to convey a clarification or change to the original drawings issued.

Addenda: Similar to the Bulletin but released by the Architect prior to the signing of a contract between the Owner/Architect and the Construction Team.

Drawing: A 2D representation of a collection of objects that are contained within the model. This may be seen as a view of the model in 2D for a select number of objects within a View Type (such as plan or section).

Specification: A written representation of the objects within the model with instructions on how they are to be constructed (such as materials to be used, techniques in construction, show drawings to be submitted, etc.)

References: Any pertinent references or background materials used

None at this writing.

Contributors: The names and chapters of the domain participants

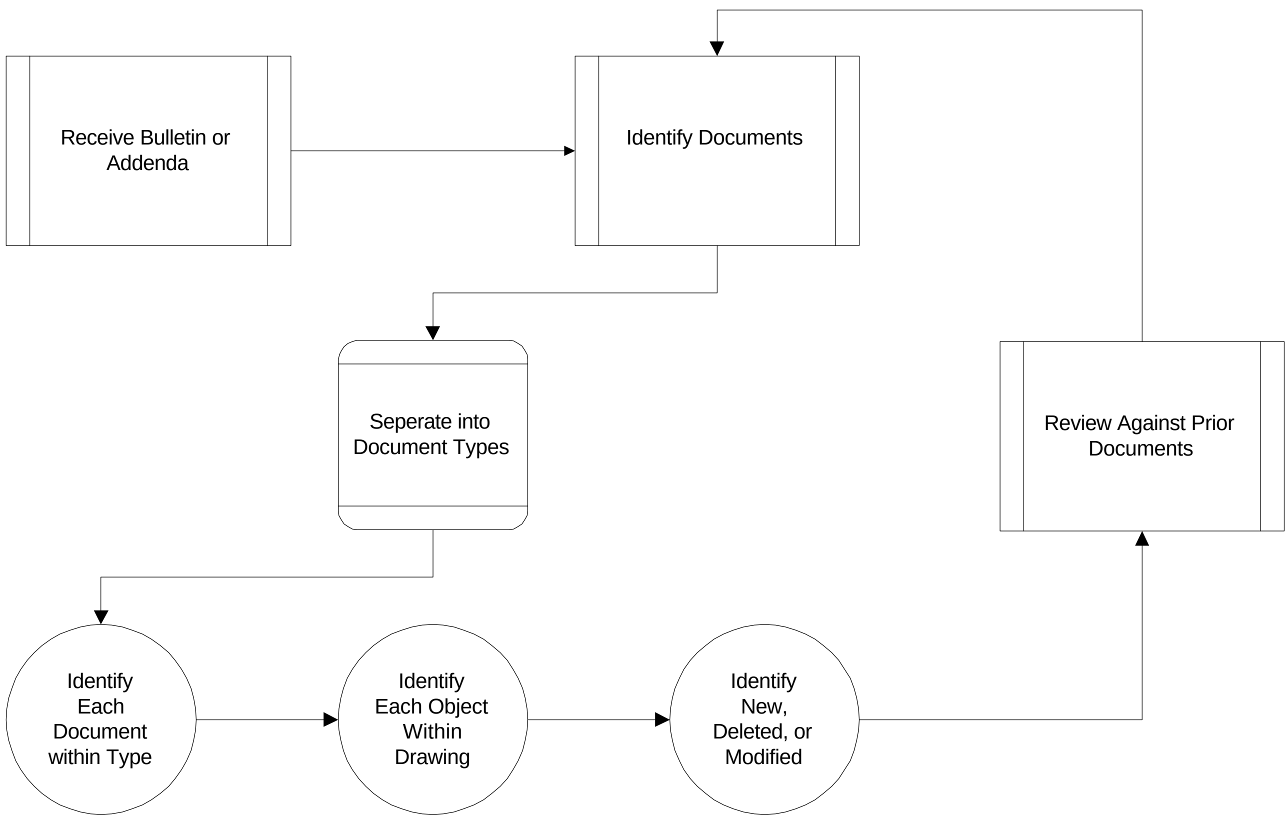
Raymond H. Brungard	NA
Graham Storer	UK

Arto Kiminieri	Nordic
Richard See (temp)	NA
Ken Herold (part time)	NA
Mike Cole (part time)	NA

3.24.1.2 Process Diagram

The diagram should illustrate how the tasks use the model to get data that was created by previous processes and to store data that may be used by later processes.

See Process Diagrams METAD1.vsd and METAD2.vsd under separate cover.



3.24.1.3 Usage Scenario

3.24.1.3.1 Overview

This section should include overview information about subject process. This overview provides a sentence or two about each bubble in the Process Diagram. The overview should conclude with what bottlenecks or areas of difficulty are frequently encountered in this process, and the benefits of enabling this process through IFC's.

The basic requirement of this process is to be able to create and retrieve views of the model which relate to the objects as 2D drawings used to. This means that a selection of objects may be chosen with a view type (the way in which the objects are to be viewed in 2D, such as plan or section view) to represent a discrete area or areas within the project. These areas can be interpreted as drawings in the sense that they may be printed out or viewed in the same manner as drawings are used today.

Process Task Descriptions

{An item by item discussion of each bubble in the Process Diagram. This should be as verbose as possible, with a clear distinction between what is desirable in a potential software application and the discreet sub-tasks required for the process. The following is a list of suggestions and recommendations for completing these sections.

Use graphics whenever possible, but be cautious with graphics that are either proprietary or too detailed to be legible at small scale.

Units can be referenced where appropriate, but do not use specific unit types to help describe the process.

Good: "The engineer calculates the room area (i.e., square feet or square metres)."

Bad: "The engineer calculates the number of square feet in the room."

Make as many references to specific attributes as possible, but do not try to define how the attributes will be modeled.

Good: "The solar transmittance is required to complete the calculations"

Bad: "The solar transmittance (Attribute name: SolarTransmittance: Data type: IfcReal, Object Class: IfcWindow) is required to complete the calculations").

Where industry-specific acronyms are required, make sure they are defined in the Definitions section of the Introduction.}

3.24.1.3.2 Task 1 - Create Drawing View:

Identify Objects within the model to include in the Drawing View. These objects should be a complete representation of the work for its' view.

Identify the view type used to represent the objects within the drawing. This view type represents the way in which the objects are viewed, usually representing a direction of view, such as plan (viewing from the top).

Provide and apply a reference number, name, revision number, and general information regarding the intended drawing.

Provide for drawing "types", such as plumbing, electrical, concrete, etc.

Provide for additional references for aggregation of information such as Bulletin, Addenda, etc.

3.24.1.3.3 Task 1 - Retrieve Drawing View:

Receive the Bulletin, Addenda, or drawing set and their references.

Identify the Drawings within the set.

Identify Drawing type.

Identify the Objects within the drawings

Identify the View of the Objects.

Retrieve additional references.

3.25 XM-3 IFC Model - Enabling Mechanisms

{{ Process definition and Usage Scenaria for this project not yet available }}

*** Instructions for completing AEC Process Definitions***

Writing "AEC Industry Process Definitions" documents

This document is provided as a formatting template for the documentation of the AEC industry processes your project team is proposing be supported by the IFC Release 2.0 project model as well as a proposed reference usage scenario for each process step. These usage scenaria should include real AEC examples (including graphics) to insure that prospective implementers understand your requirements.

The document is structured so that several process definitions can be combined into a single integrated document for all R2.0 projects.

Therefore ***please do not modify the document structure***.

Document structure is roughly:

Heading 1	1.	Project Document TYPE header
Heading 2		Industry Domain
Heading 3	1.1	Project ID and Name
Heading 4	1.1.1	Process Name
Heading 5	1.1.1.3	Intro, Process Diag., Process Requirements Definition
Heading 6	1.1.1.3.1	Process Step - detailed definition and example usage scenario

Each process analysis is presented within a “Heading 4” block (listed as “1.1.1 {{ Process/Functionality Name }} “). Below this, within each "Heading 5" block (listed as "1.1.1.1 {{Process step A}}"), a process step is defined and then analyzed in terms of its required input and resulting output information.

Notes:

1. Anything text that is *italicized* and enclosed in double curly braces → *{{ xxx }}* ← are notes to you (the writer) → should be removed when you have filled in the section.
2. Anything text that is not italicized and enclosed in double curly braces → {{ xxx }} ← are placeholders for the text you will insert. You should replace these with content as described in the instructions.
3. We will be using the TQM diagramming template in Visio for the process diagrams. Some other process definition methods are still under consideration for the longer term process definitions, but we must use what we understand at this point.
   Note that you should include all of the primary steps in your process and identify the informational requirements for each step. Details for these informational inputs/outputs will be described in the companion project document → IFC Model Requirements Analysis.
   A sample Visio 4.0 TQM diagram is included → just double-click to edit
4. Please see the example file sent with this template → R2pr_EX1.doc.

5. Naming convention for your "AEC Industry Processes and Usage Scenaria" -- R2pr_XXn.doc (where XXn is the Proj ID in the first column of the spreadsheet). Example: "Completion of the Architectural Model" has an ID of AR-1 --- thus the filename will be "R2pr_AR1.doc".
6. This document will become part of Volume I for the IFC Release 2.0 Specifications, which will be entitled → AEC Processes Supported by IFC R2.0.
7. Analysis of the informational (and behavioral) requirements for object definitions in (or to be added to) the IFC model is the focus of the companion project document → IFC Model Requirements Analysis. That document will reference the process steps defined here.
8. The process definition sections from this document must be consistent with those in the "IFC Model Requirements" document for your project.
9. This template extends previous versions of the template for this document in multiple ways, including:
    - *AEC Domain groupings have been added as "Heading 2" which demotes previous headings 2, 3, and 4*

If you have questions please send email!

Regards,

Richard See

3.26 Process Definitions Template {{Project ID - Project Name }}

Processes Defined in this project: *{{ a list of the AEC processes to be supported. }}*

1. {{ Process Name }}
2. {{ Process Name }}

3.26.1 {{Process Name}}

3.26.1.1 Introduction

Overview: *{{ A couple of sentences about this process. }}*
{{ introduction/overview to/for this process definition }}

Process Scope: *{{ An itemization of the process steps that are within scope for this process (and detailed in the process diagram and Usage Scenario below). }}*

- {{ item 1 }}
- {{ item 2 }}

Out-of-Scope: *{{ An itemization of any process steps (or sub-processes) that have been purposely omitted from this process definition and requirements }}*

- {{ item 1 }}

Definitions: *{{ Any industry-specific definitions or acronyms that are used in this section }}*

- {{ Term 1 - definition }}

References: *{{ Any pertinent references or background materials used }}*

- {{ reference 1 }}
- {{ reference 2 }}

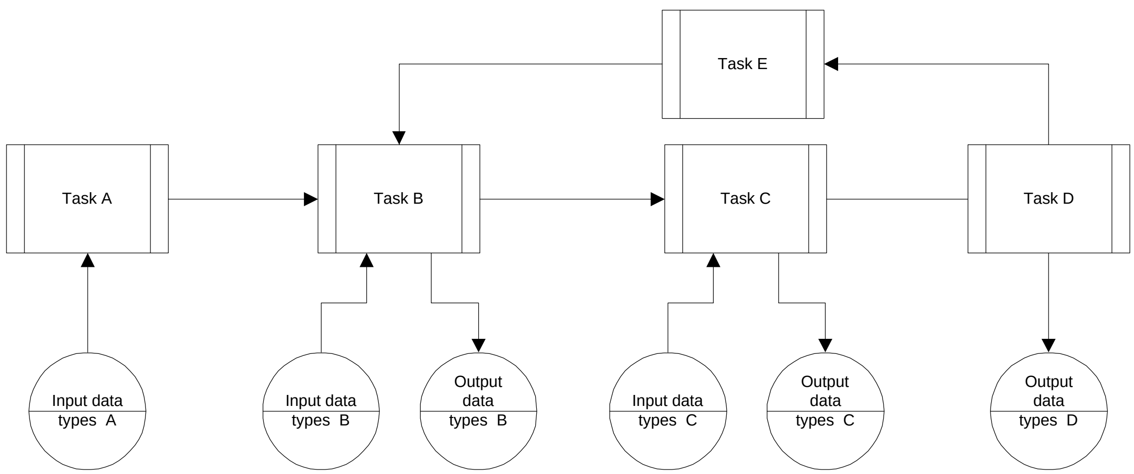
Contributors: *{{ The names and chapters of the domain participants }}*

- {{ contributor 1 }}
- {{ contributor 2 }}

3.26.1.2 Process Diagram

{{ Visio TQM Process Diagram. This should represent an exploded view of a bubble on an overall domain process diagram (the reference Industry Process Framework in the Spec Volume I - "AEC Processes to be Supported by IFC R2.0").

The diagram should illustrate how the tasks use the model to get data that was created by previous processes and to store data that may be used by later processes. These information inputs and outputs will be fully analyzed in another document à "IFC Model Requirements Analysis" }}



3.26.1.3 Process Definition

3.26.1.3.1 Overview

<< Overview description >>

{{ This section should include overview information about the subject process. This overview does not need to include each process step (which are defined below with example usage scenaria), but should give the read a general understanding of the process and how it relates to other processes.

The overview should conclude with what bottlenecks or areas of difficulty are frequently encountered in this process, and the benefits of enabling this process through IFC's.

What follows below is an item by item discussion of each task bubble in the Process Diagram. This should be as specific as possible, with a clear distinction between what is desirable in a potential software application and the discreet process tasks required for the process. The following is a list of suggestions and recommendations for completing these sections.

Use graphics in the Example Usage Scenario secitons whenever possible, but be cautious with graphics that are either proprietary or too detailed to be legible at small scale.

Units can be referenced where appropriate, but do not use specific unit types to help describe the process.

Good: "The engineer calculates the room area (i.e., square feet or square metres)."

Bad: "The engineer calculates the number of square feet in the room."

Make as many references to specific attributes as possible, but do not try to define how the attributes will be modeled.

Good: “The solar transmittance is required to complete the calculations”

Bad: “The solar transmittance (Attribute name: SolarTransmittance: Data type: IfcReal, Object Class: IfcWindow) is required to complete the calculations”).

Where industry-specific acronyms are required, make sure they are defined in the Definitions section of the Introduction *}}*

3.26.1.3.2 {{ Process Task A }}

Task Description: *{{ thorough description of the task }}*

{{ Description }}

Example Usage Scenario: *{{ an example usage scenario for completing this process using data, graphics, etc. from a real AEC project. Try to insure that you have a single project example that spans the full process so that your examples flow from one process task definition to the next. }}*

{{ example scenario description, example scenario description, example scenario description, example scenario description, example scenario description, example scenario description, example scenario description, example scenario description, example scenario description, example scenario description, example scenario description, example scenario description. }}

Sample Graphics

3.26.1.3.3 {{ Process Task B }}

Task Description: *{{ thorough description of the task }}*

{{ Description }}

Example Usage Scenario: *{{ an example usage scenario for completing this process using data, graphics, etc. from a real AEC project. Try to insure that you have a single project example that spans the full process so that your examples flow from one process task definition to the next. }}*

{{ example scenario description, example scenario description, example scenario description, example scenario description, example scenario description, example scenario description, example scenario description, example scenario description, example scenario description, example scenario description, example scenario description, example scenario description. }}

Sample Graphics

4. IFC Model Requirements Analysis

Architecture

4.1 AR-1 Completion of the Architecture Model

4.1.1 Building Shell Design

The architect balances the building massing with the elevation aesthetics while performing exterior shell design. Both processes (massing and shell design) evolve and cycle back and forth as each may change aspects of the other. The exterior shell design involves making the massing interesting while using glass fenestration, cladding materials, and details in adornment that create a scale and design motif. Other aspects of this process, that are balanced, are the need for visual access and illumination of the spaces behind the shell, and the issues of attaching and waterproofing the shell. The shell design starts typically after a preliminary space layout and during the building massing studies.

4.1.1.1 Introduction

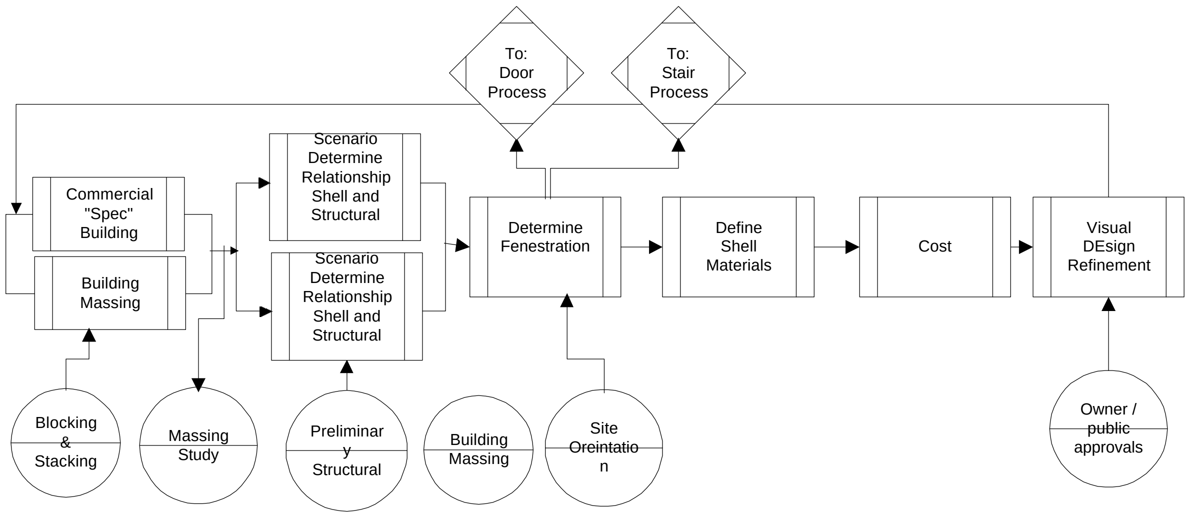
The architect starts the shell design by working with the preliminary stacking and blocking diagrams to determine a massing of the building, based on the floor plates created in the space layout phase. After the massing, the architect will determine the proper aesthetics effect for the building, whether the facade is connected to the outside of the structure or integrated within the structure. The fenestration is determined based on the amount of light and visual impact of the glass and openings on the facade. After the designer determines the type of materials used, preliminary heat gain/heat loss can be calculated for operational cost impact of the building shell. With the final selection of material and fenestration, a detailed design of the adornment of the facade proceeds using reveals, treatment of the materials, cornices, and other building design elements.

Out-of-Scope*: block and stacking, site analysis and location of the building.*

Definitions:

- *Shell - The exterior wall of a building. Other terms used (facade, elevation, building envelope)*
- *Massing - The exterior shape of a building. A volumetric view of the building.*

4.1.1.2 Process Diagram



4.1.1.3 Process Analysis

Building Massing (option 1)

The preliminary building massing takes the block and stack, along with a preliminary structural plan, and defines the volume of the building and a floor plate shape.

Input Information:

- *A bubble diagram laid out floor by floor (Architecture block & stack)*
- *Structural depths (Structural)*
- *Preliminary HVAC depths (HVAC)*

Output Information:

- *Refined floor plate shapes (Structural, Architecture)*
- *Refined floor to floor heights (Structural, Architecture)*
- *Volume and massing of the building (Architecture, HVAC, Simulation, Analysis)*
- *Preliminary elevation shape (Architecture)*
- *Exterior Circulation (ramps, balconies, docks, stairs, elevators)*

Commercial Spec Building - Based on Buildable Area (option 2)

Input Information:

- *Buildable footprint from site analysis*
- *Client Progam and Budget for building square footage, efficiencies, and image*
- *Structural depths (Structural)*
- *Preliminary HVAC depths (HVAC)*

Output Information:

- *Refined floor plate shapes (Structural, Architecture)*
- *Refined floor to floor heights (Structural, Architecture)*
- *Volume and massing of the building (Architecture, HVAC, Simulation, Analysis)*
- *Preliminary elevation shape (Architecture)*
- *Exterior Circulation (ramps, balconies, docks, stairs, elevators)*

Determine Relationship between Shell and Structure

The relationship of the shell and structure is defined based on the effect the architect wants to achieve with the design. For example, the shell may be attached to an edge of slab and column which are flush. On the other hand, the columns may protrude through the shell creating a facade that expresses the structure.

Input Information:
- *Preliminary Massing Studies*
- *Preliminary Design Grid (Architecture)*
- *Preliminary Structural Grid/System (Structural)*

Output Information:
- *Floor plates and design grid (Structural, Architecture)*
- *Refined elevation and model (Architecture)*
- *HVAC system (volume, simulation)*

Determine Fenestration (aesthetic criteria)

The determination on the fenestration is based on the rhythm and effect the facade should have with respect to glass area. At this stage, a decision on the type of window is made but not detailed. Examples of this process is to determine whether windows are punch into a facade or are flush.

Input Information:
- *Refined floor plate shapes (Structural, Architecture, Construction)*
- *Refined floor to floor heights (Structural, Architecture)*
- *Preliminary Structural Depths (Structural)*
- *Volume and massing of the building (Architecture, HVAC, Simulation, Analysis)*
- *Code requirements (fire access, sill heights, energy)*
- *Preliminary elevation shape (Architecture, Structural)*
- *Building Orientation (Architecture)*

Output Information:
- *Window/Door dimensions (Architecture, HVAC, Simulation, Construction, Analysis)*
- *Window/Door locations (Architecture, HVAC, Simulation, Construction, Analysis)*
- *Glass Area (Architecture, HVAC, Simulation, Construction, Analysis)*
- *Window/Door Type (Architecture (HVAC, Simulation, Construction, Analysis)*
- *Window/Door Framing (Architecture, HVAC, Simulation, Construction, Analysis)*
- *Shading elements (overhang, brise desoleil, landscape elements, Analysis)*

Define Shell Materials

A decision is made on the types of material that will be used for the shell. This will effect how the facade materials will be attached to the structure.

Input Information:
- *Project Material List (Architecture, Client)*
- *Construction Methods (Construction)*
- *Code Considerations*

Output Information:
- *Exterior wall type (HVAC, Simulation, Structural, Construction, Analysis)*

Composition
Materials

Connections
- *Window/Door Type*

Composition
Materials
- *Project documents (information to others)*

Costs

A preliminary analysis may be run to determine the effect of the shell design on the energy used during operation of the building as well as energy consumed to construct the building.

Input Information:
- *Fenestration (Architecture)*
- *Wall type (Architecture)*
- *Window/Door type (Architecture)*
- *Exterior Circulation (ramps, balconies, docks, stairs, elevators)*
- *Preliminary HVAC system*
- *Occupancy*
- *Loads (lighting, ventilation)*
- *Waste Stream (greening)*

Output Information:
- *Heat gain numbers*
- *Heat Loss numbers*
- *Preliminary energy analysis*
- *Material*
- *Equipment*
- *Life Cycle Costs/Trade-Offs*
- *Waste Stream/Trade-Offs (greening)*
- *Construction Time*

Visual Design Refinements

At this point in the process, the shell is refined and detailed. This may include finishes, additions or treatment to materials such as flame/rough/polished stone, reveals, setting back panels, cornices, or parapets. Each of the adornments are used to add character to the design of the facade.

Input Information:
- *Wall Type (Architecture)*
- *Design Character/Adornment (Architecture)*

Output Information:
- *Details on adornment (Structural, Construction)*

4.1.1.4 IFC Model Impact

New object types *(need to know: floor to floor; floor plate; topography(grade))*

- *Parapet*
  - *Height*
  - *Width*
  - *CapDetail (assembly of cap)*
  - *Material 1*
  - *Material 2*
  - *Material n*
- *Louver*
  - *Height*
  - *Width*
  - *Material*
  - *Finish*
  - *screen/mess size*
- *Stair (See Stair Process)*
- *Ramp*
  - *Width*
  - *Slope*
  - *Material*
  - *Finish)*
  - *Handrail*
- *Canopy*
  - *Height*
  - *Slope*
  - *Materials*
  - *Finishes*
  - *Depth*
  - *{{ attribute 1 }}*
- *Projections (ornamentation)*
  - *Type*
  - *Length*
  - *Material*
  - *Weight*
  - *Orientation (vertical, horizontal, etc)*
  - *Connection (connection to facade ie. bolt, steel)*
- *Expansion joints*
  - *Width*
  - *Material*
  - *Location (Polycurve)*
  - *{{ attribute 1 }}*
- *Guardrail*
  - ***(See Stair Process)***
- *Curtain wall (window wall)*

  - *{{ attribute 1 }}*
  - *{{ attribute 1 }}*
  - *{{ attribute 1 }}*

- *Foundation (elements, connections) see foundation design*
- *Roof (connections) see roof design*
- *Spandrel*
  - *Width*
  - *Material (assembly)*
  - *Connector*
- *Material attribute*
  - *Weatherability (durability) life cycle*

Extensions to R1.0 object types

- *Window*
  - *{{ attribute 1 }}*
- *Exterior door*

  {{ attribute 1 }}

4.1.1.5 RoadMap Issues

Interoperability issues

Disciplines from which information is needed:

- *Structural*
- *HVAC*
- *Energy*
- *Codes*

Disciplines for which information is produced:

- *HVAC*
- *Simulation*
- *Construction*
- *Facility Management*
- *Specifications*

4.1.2 Building Core Design

The core design is a balance between making available ancillary spaces and program requirement. The size and location on a floor is determined by the structural systems, program requirements including number of occupants and building codes such as ADA. The design of the core follows the initial layout of the spaces defined in the building program. The spaces that make up the core are typically not defined in the program but are extracted by information about the floor size and occupants.

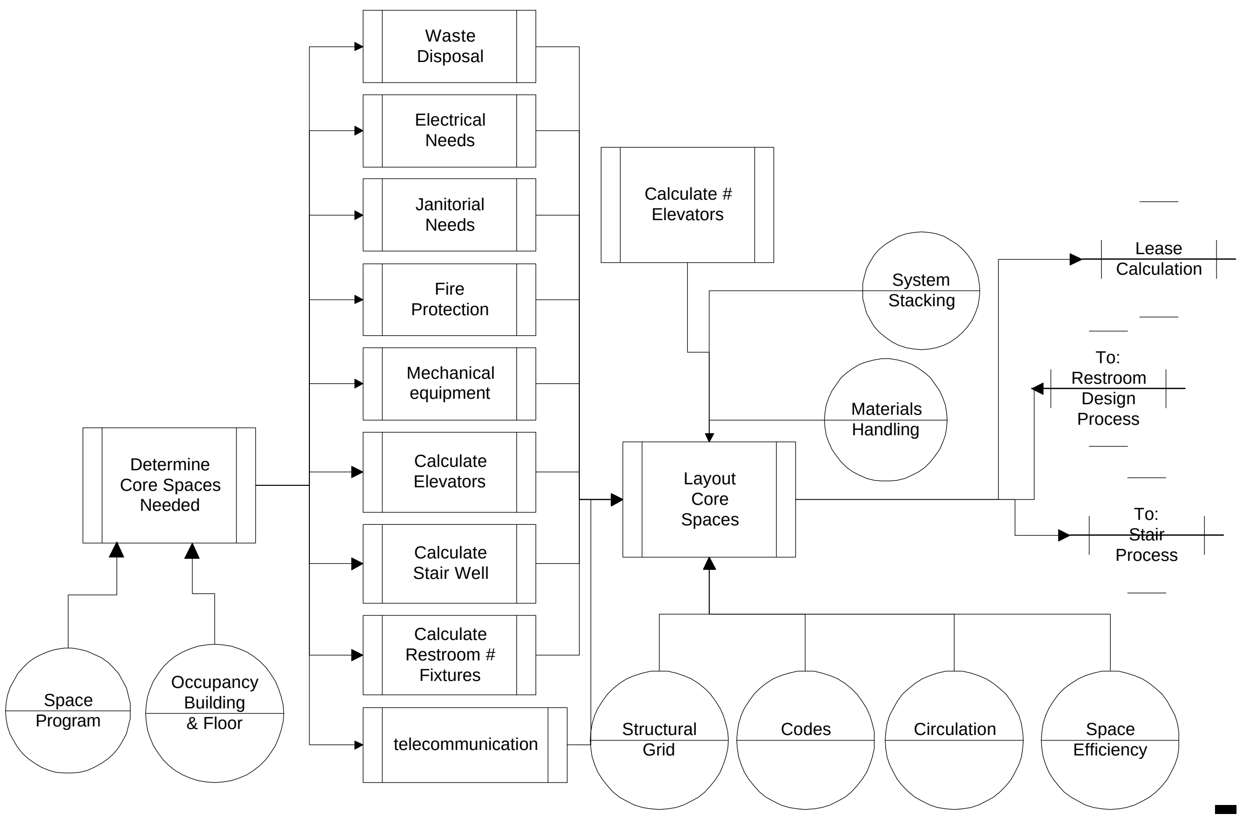
4.1.2.1 Introduction

The core design starts by determining the size of the items needed in the core. Calculations for the number of elevators are based on building occupants and number of floors. The restroom size is based on the number of occupants on the floor and in the building. The floor to floor height is used to determine the length of the stairs which determines the size of the stairwell. The circulation around the core is determined by the type of occupancy and fire codes. The layout of the pieces of the core are driven by the structural grid and distances determined by codes, etc.

Process Scope: Assumptions /presumptions: space program (owners' criteria); occupancy, building, floor; parking garage impacts (structural grids); materials handling (site delivery, building services). The core is defined as items for circulation and service delivery for occupants. It does not have to be in the center of the building.

Out of Scope: This process does not address the actual design of stairs, restrooms, parking design and lobby design. Also materials handling and entering and exiting the building are not included in the core design.

4.1.2.2 Process Diagram



4.1.2.3 Process Analysis

Determine Core Spaces Needed

Determine the type and number of spaces to be included in .

Input Information:

- *Space program (owner requirements)*
- *Occupancy (Floor by Floor)*
- *Occupancy Type (Assembly, etc in code)*
- *Codes/Egress (Distances) (Look to AR-2)*
- *Building Services (# and type of service)*
- *Vertical Circulation (#, type)*

Output Information:

- *Spaces (#,type)*

Determine Core Space Sizes

Apply codes and other processes to determine the size and shape of core spaces.

Input Information:

- *Calculate elevators (Number and sizes length, width, and type (freight vs. passenger)(Freight lobbies)*
- *Calculate Stairs (process #### Length, width)*
- *Floor to Floor Heights*
- *Number of Floors*
- *Calculate Escalator (width, length)*
- *Alarm Stations (width, length)*
- *Restroom Design (process #### length, width, area)*
- *Required spaces (length/width or area) Electrical, Communications, Waste Disposal, Janitorial, Mechanical*

Output Information:

Required spaces (length/width or area) (Collection of spaces ie (Core, parking)

Layout Core Spaces

Layout core spaces.

Input Information:

- *Structural Grid (grid of object, including shear walls etc.*
- *Max. Distance between exit stairs.*
- *Space efficiency (% usable goal)*
- *Parking Plan*
- *Required spaces (length/width or area) Electrical, Communications, Waste Disposal, Janitorial, Mechanical, Stair, Elevator, Escalator*

Output Information:

- *Core layout (collection of spaces)*

Detailed Design of Stairs

Covered in Stair design Process.

Detailed Design of Restrooms

Covered in Restroom design Process.

4.1.2.4 IFC Model Impact

New object types

- *Stairs (Actual object*
  - *{{ attribute 1 }}*
- *Stairs Well*
  - *{{ attribute 1 }}*

- *Elevator Shaft*
  - *{{ attribute 1 }}*
- *Elevator*
  - *{{ attribute 1 }}*

- *Escalator Shaft*
  - *{{ attribute 1 }}*
- *Restroom*
  - *{{ attribute 1 }}*
- *Storage*
  - *{{ attribute 1 }}*
- *Electrical closet*
  - *{{ attribute 1 }}*
- *Telecommunications closet*
  - *{{ attribute 1 }}*
- *Mechanical closet*
  - *{{ attribute 1 }}*
- *Janitorial closet*
  - *{{ attribute 1 }}*
- *Emergency services*
  - *Fire Standpipe*
  - *Hose*
- *Circulation*
  - *{{ attribute 1 }}*
- *Refuse Area*
  - *{{ attribute 1 }}*
- *Lobby*
  - *{{ attribute 1 }}*
- *Chase*
  - *{{ attribute 1 }}*

Extensions to R1.0 object types

- *{{ Object type name }}*
  - *{{ attribute 1 }}*
  - *{{ attribute 2 }}*

- *{{ Object type name }}*
  - *{{ attribute 1 }}*
  - *{{ attribute 2 }}*

4.1.2.5 RoadMap Issues

Interoperability issues

Disciplines from which information is needed:

- *Structural*
- *HVAC*
- *Telecommunications*
- *Plumbing*
- *Electrical*

Disciplines for which information is produced:

- *Structural*
- *HVAC*
- *Telecommunications*
- *Plumbing*
- *Electrical*
- *Specifications*

4.1.3 Roof Design

The process of roof design is a mixture of aesthetics, weather dissipation, and hiding other building objects such as telecommunications, mechanical, and elevators. The process is iterative, the designer works back and forth between the massing and roof design to create a building design which expresses a character appropriate to the area, client wishes, and building type.

4.1.3.1 Introduction

The architect determines a type of roof based on the design direction and the character of the building. Using the building massing, the architect lays out the roof. On pitched roofs, refinement of the intersection of the roof planes will be necessary. The architect then determines and designs the drainage. The intersection of the roof with the elevations are designed and detailed. The layout and penetration of other services that are hosted on the roof are considered. Materials are selected.

Definitions:

Dormers (space projection from sloped roof, may be considered standard roof, not unique)

Recreation areas

Helipads

Steeples can also be used as a screen or just ornate

Screening

Chimneys

Vents

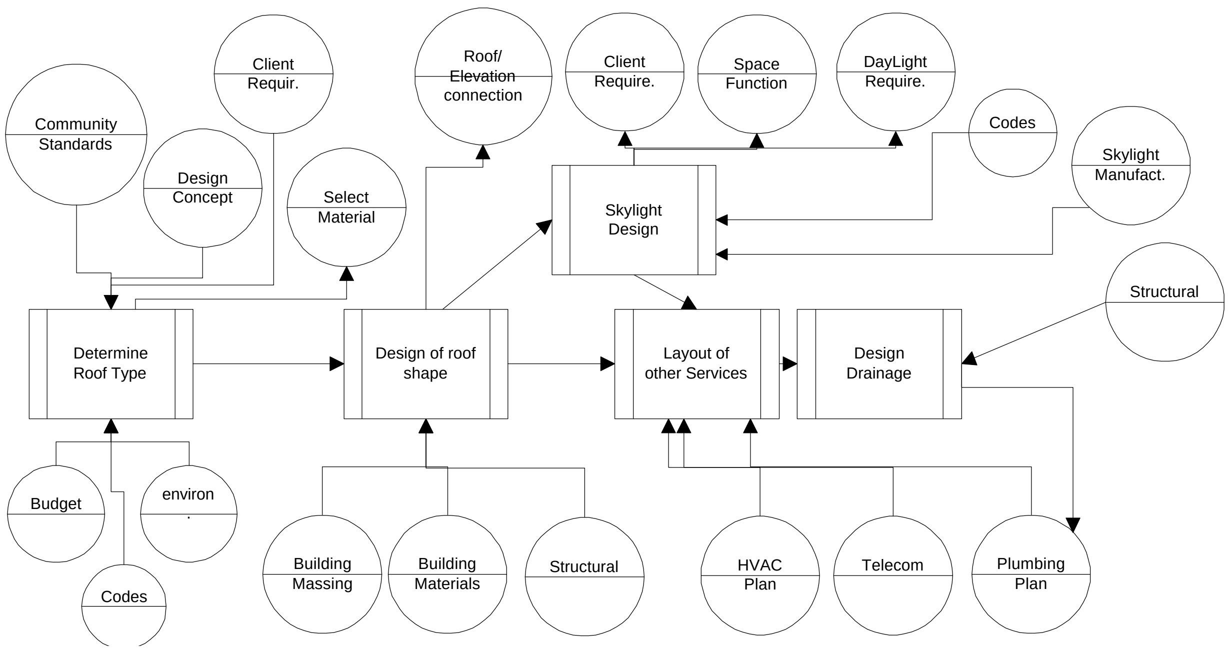
Drainage

Telecommunications: Transmission Tour

Process Scope: Design inputs would cover the process of exterior and interior programs including eaves and overhangs. Interior issues need to address cathedral ceilings, dormers, etc. Exterior roof issues include steeples, parapet roof ventilation, electrical, drainage, recreational areas, planters, irrigation, window washing, skylights, smoke evacuators, access hatches, mechanical screens, roof walk pads, lighting control, and FAA lighting.

Out of Scope: Actual design of electrical, venting, access hatches, smoke evacuators, sidewalk protection canopies.

4.1.3.2 Process Diagram



4.1.3.3 Process Analysis

Determine Roof Type

The process of deciding the type of roof is a mixture of balancing the form of the building with the local style and the desire of the client. The roof type refers to flat, pitched, gabled, etc.

Input Information:

- *Budget constraints*
- *Community and regional standards*
- *Environment such as snow or tepid regions*
- *Design intent (hiding building services)*
- *Client Requirements*
- *Functional requirement (structural loading)*

Output Information:

- *Basic form of roof (i.e. Flat, pitched, shed, etc.)*

- *Material requirements (i.e. clay tile roofing, slate)*

Design Roof Shapes

After the selection of the roof type, a preliminary design is produced to determine the actual shape and its impact on the building form.

Input Information:

- *Codes (fire, class, slopes)*
- *Building massing*
- *Building materials*
- *Structural*
- *Cost*
- *Surrounding Building Scapes*

Output Information:

- *Slope*
- *Structural*
- *Area of roof planes*
- *Shade shadow*
- *Vert/horz projections*
- *Skylight locations*

Skylight/Clear Story

After the shape is created, the integration of any skylights or clear story windows will be integrated into the roof to evaluate the impact and location based on preliminary structural ideas.

Input Information:

- *Codes*
- *Client requirements*
- *Day lighting*
- *Design Intent*
- *Energy requirements*
- *Manufacturer input*

Output Information:

- *Slope of Skylight*
- *Glazing area*
- *Materials*

Layout of Services

With the major shape and items such as skylights, etc. determined, the architect then looks at the projections through the roof of items such as vents, stair/elevator, and mechanical.

Input Information:

- *HVAC equipment and piping locations*
- *Telecommunications needs in respect to roof dishes etc.*
- *Plumbing venting stacks*
- *Circulation*
- *Fire Protection*

Output Information:
- *Location of services*
- *Roof plan*
- *Walkways/Roof Access*
- *Building maintenance equipment (window washing)*
- *Heliport*
- *Health and Fitness (pool, tennis courts, 5pm)*

Design Rain/Snow Drainage

At this point, the runoff of water is calculated and a design concept is created to use roof drains, scuppers, or gutters.

Input Information:
- *Structure*
- *Roof plan*
- *Geographic location and weather information.*

Output Information:
- *Water/Snow drainage plan*
- *Rough drain/downspouts location and sizes (interior drainage)*
- *Design Scupper*

4.1.3.4 IFC Model Impact

New object types

- *Roof*
  Type (flat, sloped)
  Material (assembly)
  Classification (A,B)
- *Skylights Opening (could be domed, barrel vault)*
  - *Width*
    Length
    Height
  - *GlazingType*
  - *WaterProofMethod (assembly)*
- *Scupper*
  - *Width*
  - *Depth*
  - *Location*
  - *Material*
- *Drainage gutters*
  - *Material*
- *Mech screen*
  Length
  Width
  - *Height*
  - *Type (assembly)*
- *Window cleaning (rigging, tracks, rails, carriage, apparatus, maybe this should be pulled out as a process)*
  - *Location*

   - *Type*
   - *Connection*
- *Dormers*
   - *Shape (polygon surfaces)*
   - *WindowType*
   - *Location*
   - *Material*
- *Chimneys*
   - *Length*
   - *Height*
   - *Composition (assembly)*
   - *Width*
   - *Flue*
   - *Cap*

- *Projections*
   - *Type*
   - *Length*
   - *Material*
   - *Weight*
   - *Orientation (vertical, horizontal, etc)*
   - *Connection (connection to facade ie. bolt, steel)*
- *Hatches*
   - *Length*
   - *Height*
   - *Composition (assembly)*

     Width

- *Parapet (ken, ask hok dudes)*
-
- *Stairs (See Stair Process)*
- *{{ attribute 1 }}*
- *Vents*
   - *Length*
   - *Height*
   - *Composition (assembly)*
   - *Width*
- *Walkways*
   - *Material*
   - *Path*
   - *Composition (assembly)*
   - *Width*

Extensions to R1.0 object types

- *{{ Object type name }}*
   - *{{ attribute 1 }}*
   - *{{ attribute 2 }}*

- *{{ Object type name }}*
- *{{ attribute 1 }}*

- *{{ attribute 2 }}*

4.1.3.5 RoadMap Issues

Interoperability issues

Disciplines from which information is needed:

- *Structural*
- *HVAC*
- *Plumbing*
- *Telecommunications*
- *Electrical*
- *Municipal codes*

Disciplines for which information is produced:

- *Structural*
- *Plumbing*
- *Telecommunications*
- *HVAC (heat gain/heat loss analysis)*
- *Electrical*
- *Municipal Codes*
- *Specifications*

4.1.4 Blocking and Stacking

4.1.4.1 Introduction

Overview: Space (area or volume) blocking and stacking is a process of converting the organizational needs of a client into a graphic description of location of spaces and their relationships. After the areas/volumes (bubble diagrams) are created, the designer places them horizontally(blocking) and vertically (stacking).

Process Scope: The scope of this process encompasses: defining spaces, naming of areas/volumes, calculation of spaces, checking mechanisms (max/min standards, confirming design against criteria), establishing adjacencies, grouping spaces, circulation (basic load factoring, % assignment), defining organizational structures, design criteria (site analysis (criteria), grid office layout, day lighting) and existing conditions. Cost referencing or relative quality of spaces (FM reference) may be included. (Need to coordinate with FM domain group)

Out-of-Scope: This process does not intend to address programming, escape (egress) code analysis, or budgeting.

Definitions: Blocking - horizontal placement of spaces

Stacking- vertical placement of spaces

Adjacency - relationship of spaces (both horizontal and vertical)

Space

Grouping, biz-unit, department /relationship to space/adjacency

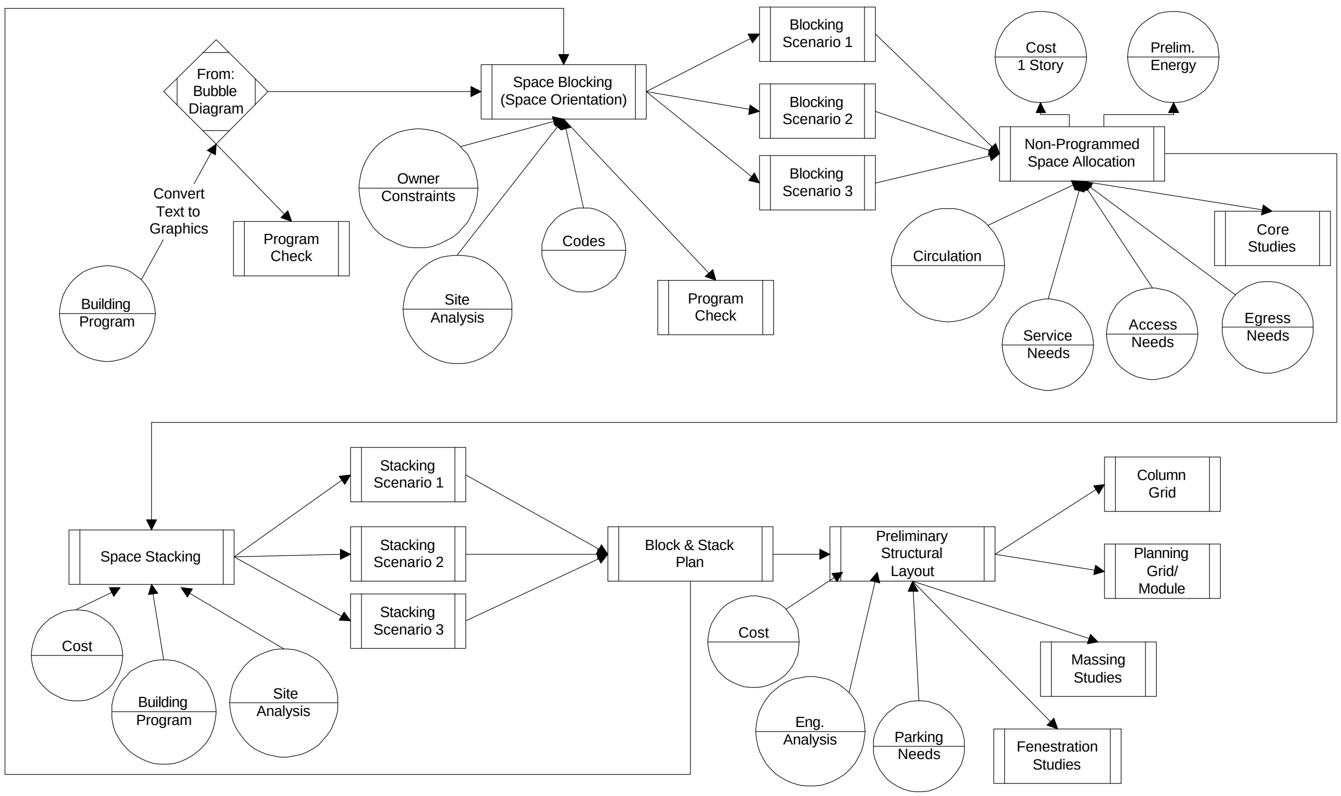
Programmed space

Non-programmed

References: Historical Projects, Project Libraries, Corporate Databases

Contributors: NA, D, UK, No, Sg

4.1.4.2 Process Diagram



4.1.4.3 Process Analysis

Generate Bubbles

Reference Version 1.0

Input Information:

- *Building Program (Client)*

Output Information:

- *Program Check*
- *Preliminary layout of spaces*

Non-Programmed Space Allocation

Many programs do not define explicitly the size of circulation spaces and the core spaces such as bathrooms, stairs, and elevators. In many cases, a percentage of occupied space is used to determine the amount of space for circulation.

Input Information:

- *Owner Requirements (Circulation %, Quality(cost))*
- *Ancillary Space (area) (service circulation, utility ())*
- *Codes (egress, Health code adjacency)*

Output Information:

- *Bubbles Non-Programmed Spaces*

Space Blocking/Stocking

Arranging space bubbles in the plan, while using the building program to provide adjacency information, will help indicate which spaces should be placed next to others. For a building that is multi-story, the designer will stack the grouped spaces vertically on floors. Stacking may involve splitting spaces that do not fit totally on one floor to other floors.

Input Information:

- *Site Analysis (view (orientation), buildable area , solar(orientation), access (delivery, pedestrian, emergency, utilities (power, etc))*
- *Building Footprint (polygon)*
- *Parking (stall/sqarea, numb of stalls(compact, large, standard, handicapped, Van/car pool stalls), clearance)*
- *Owner Constraints (program)*
- *Municipal Constraints (shear wall, fire wall, incentives (sqarea))*
- *Public Constraints (pedestrian requirements (vector))*

Output Information:

- *Building Footprint*
- *Blocking Scenarios (Plans)*
- *Stacking diagrams*
- *Room Schedules*
- *Floor by Floor Area*

Preliminary Structural Layout

A preliminary indication of structural elements, such as shear walls and column grids, is needed during the process of laying out spaces.

Input Information:

- *Blocking plans*
- *Planning Grid*
- *Structural Grid*
- *Structural Type (Steel vs. concrete)*
- *Owner Constraints (Clear span needs)*

Output Information:

- *Structural Grid*
- *Building Method*
- *Footprint*
- *Floorplate*
- *Floor by Floor Area*

4.1.4.4 IFC Model Impact

New object types

- *ParkingStall*
  - *Width*
    - *Depth*
    - *Vertical Clearance*
  - *Angle*
  - *Type (compact, handicapped, normal)*
- *Parking Circulation*
    - *Width*
    - *Type (Oneway, Twoway)*
  - *EntryPath*
  - *ExitPath*
- *Ramps*
    - *Width*
    - *Length*
    - *Slope*
    - *Type (circular, straight)*
- *Adjacency (independent of groupings*
- *Building Set Backs*
- *BOMA /other standards what do we store*
- *Floor Area Ratio FAR*
    - *{{ attribute 1 }}*

Extensions to R1.0 object types

- *{{ Object type name }}*
  - *{{ attribute 1 }}*
    - *{{ attribute 2 }}*
- *{{ Object type name }}*
  - *{{ attribute 1 }}*
  - *{{ attribute 2 }}*

4.1.4.5 RoadMap Issues

Interoperability issues

Disciplines from which information is needed:

- *Owner Requirements*
- *Structural*
- *Site Planners*

Disciplines for which information is produced:

- *Architecture*
- *Structural*
- *HVAC*
- *Plumbing*
- *Electrical*
- *Construction*
- *Specifications*

4.1.5 Stair Design

Stair design is accomplished by working with the major elements, such as treads, landings, and railings, to determine the appropriate size of the stair and its elements. The process is an iterative process where the answer for one of the elements may change the size of another. The two factors that determine many of the size related decisions are based on the occupancy load and the exiting requirement.

4.1.5.1 Introduction

The architect starts the stair design by working with information about the building such a location of the stair based on egress. The width and depth is defined during a process of working back and forth. The width is determined by the number of occupants traveling through the stairwell during an emergency. The width is typically defined in the local building and fire codes. The floor to floor heights of the story are used to determine the length of the stairs, based on a rise and run. The designer may then design the depth of the landing based on codes. As the design progresses to the handrail, its design can potentially affect the width of the stairs and landing, depending on the distance it protrudes into the stairwell. At the point where the size of the treads, landing, and the handrail are set, the materials and construction methods are determined. The final design involves adding items such as exit signs, doors and hardware, and emergency lighting.

Process Scope: The process described is for fire stairs in a building. Include fire stair materials. ADA safe haven concept should be included (telecommunications, extra design space, area impact)

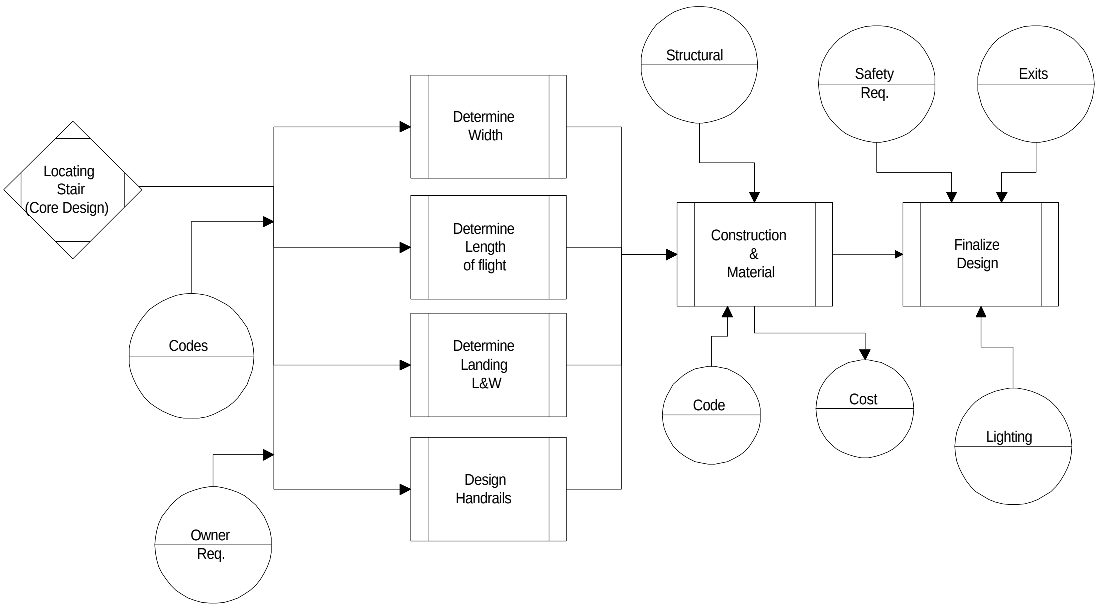
Out of Scope: Ornamental stairs not in scope and not required for exiting a floor, ladders.

Definition: ADA safe haven

References:

- Safe haven documentation
- Calculation of stair rise and run

4.1.5.2 Process Diagram



4.1.5.3 Process Analysis

Locate Stairs

The stairs are located based on the distance an occupant has to travel to exit the building in an emergency. Other factors may effect the orientation and location of the exit door, such as the core configuration, structural, and location of occupied spaces.

Input Information:

- *Configuration (type straight, Scissors)*
- *Owner Requirements*
- *Codes*
- *Occupancy*
- *Circulation*
- *Core Inputs (location exit, etc.)*

Output Information:

- *Location and Type*

Determine Width

The width of the stairs are determined by building codes which indicate the minimum sizes based on the number of occupants using the stairwell in a certain amount of time.

Input Information:

- *Configuration (type straight, Scissors, spiral)*
- *Handrail projection (Depth)*
- *Clear Area (distance) between handrail*
- *Stair use (Fire stair, Ornamental)*

- *Codes (Tread Width(distance))*
- *Egress (# of occupants by building type)*
- *Owner Requirements (Grander defined width)*

Output Information:

- *Width of treads*

Determine Tread depth and Risers height

The length of the stairs are determined by the floor to floor heights and appropriate tread depth and riser height defined in the local building code.

Input Information:

- *Floor to Floor Heights*
- *Acoustic rating (stc, impact rating)*
- *Codes (Max and min, ratio, nosing depth)*
- *Owner requirements (Depth, Rise) consistent fall within the ratio*

Output Information:

- *Tread depth*
- *Riser height*
- *Nosing Depth*
- *Landing Locations*
- *Material Type*
- *Finish*

Determine Landing

The landing width and depth is determined by stairs connected to the landing and the number of occupants switching between stair flights. The minimums are defined in the building code. ADD SAFE HAVEN

Input Information:

- *Stair Width*
- *Acoustic rating (stc, impact rating)*
- *Door, standpipe, handrail, clearance*
- *Special Criteria (depth, width)*

Output Information:

- *geometry of Landings*
- *Material Type*
- *Finish*

Handrail Design

The handrail has both an aesthetic component and is driven based on codes. When a decision is made on the type of handrail, the width of the stairs may change based on the distance the handrail protrudes from the wall.

Input Information:

- *Code (Minimum projections, min. /max height, diameter, extension from base, extension from top, continuation)*
- *Stair configuration (egress, ornamental)*
- *Special Criteria (Minimum projections, min. /max height, diameter, extension from base, extension from top, continuation)*

Output Information:
- *Handrail geometry*
- *Material Type*
- *Finish*
- *Handrail specifications (Minimum projections, min. /max height, diameter, extension from base, extension from top, continuation)*

Guardrail Design

The handrail has both an aesthetic component and is driven based on codes. When a decision is made on the type of handrail, the width of the stairs may change based on the distance the handrail protrudes from the wall.

Input Information:
- *Code (min. /max height, balustrade spacing, Minimum penetration size)*
- *Special Criteria (min. /max height, balustrade spacing, Minimum penetration size)*

Output Information:
- *Guardrail geometry*
- *Material Type*
- *Finish*
- *Guardrail specifications (min. /max height, balustrade spacing, Minimum penetration size))*

Not in scope

The final detail of stair design evolves other objects connected or part of the stair. This may include typing of exit doors, signage, standpipe location, location of vents and hatches.

Input Information:
- *Life Safety Requirements*
- *Exits*

Output Information:
- *Lighting needs*
- *Ventilation needs*
- *Pressurization*
- *Signage*
- *Stair Design*

4.1.5.4 IFC Model Impact

New object types

- *Treads*
  - *RiserHeight*
  - *TreadDepth*
  Material

    Nosings Material
    - *Type*
    - *Material*

- *Handrails*
  - *Type*
  - *Material*
  - *DepthFromWall*
- *Guardrails*
  - *Type*
  - *Material*
  - *DepthFromWall*
- *Landings*
  - *Depth*
  - *Width*
  - *Material*
- *Stringer*
  - *Depth*
  - *Width*
  - *Material*
  - *Shape (surfaces)*

Extensions to R1.0 object types

- *{{ Object type name }}*
  - *{{ attribute 1 }}*
    - *{{ attribute 2 }}*
- *{{ Object type name }}*
- *{{ attribute 1 }}*
- *{{ attribute 2 }}*

4.1.5.5 RoadMap Issues

Interoperability issues

Disciplines from which information is needed:

- *Structural*
- *Codes*

Disciplines for which information is produced:

- *Plumbing*
- *Electrical*
- *Codes*
- *Construction*
- *Facility Management*
- *Structural*
- *Specifications*

4.1.6 Restroom Design

The design of restrooms involves effective movement of building occupants, ADA codes, and aesthetic use of materials. The minimum number of fixtures is determined by the number of occupants that reside on a floor or visit a floor.

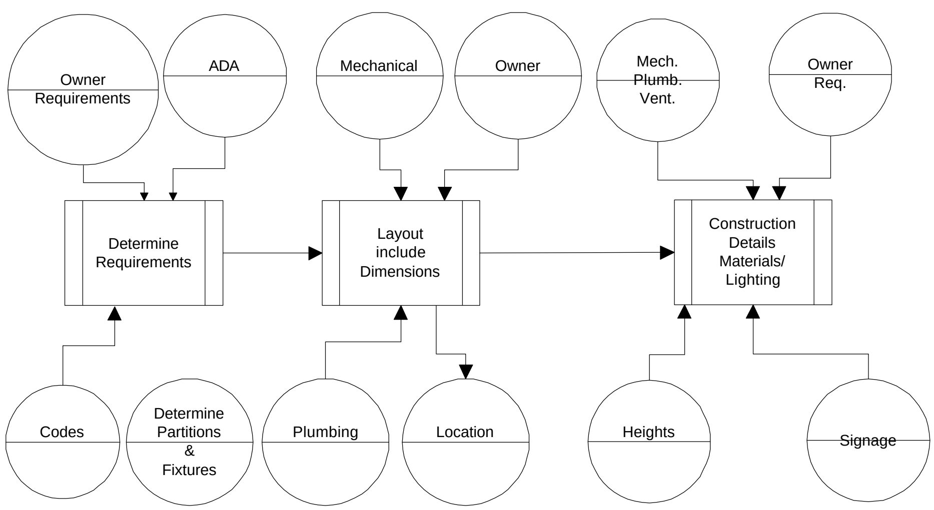
4.1.6.1 Introduction

At the start of restroom design, the number of fixtures are determined by the floor occupancy. The designer will also determine items such as partition type, fixture type, stall sizes, based on codes such as ADA and any client requirements. The next level of design involves locating the restroom fixtures and lavatories to use the most effective amount of space to contain cost but provide effective circulation. The next level of design involves locating the lavatories, mirrors, towel racks, grab bars, hand dryers, and any other object that services the restroom occupants. Appropriate location of fixtures and other items in the restroom may be determined by effective use of other building services such as plumbing stacks, etc. The final step of design is more aesthetic in that it involves the visual character of the restroom in selecting material type, sizes and objects such as faucets etc.

Out-of-Scope: Locker Rooms, Showers

In-scope: Commercial Public Restroom associated with the building core

4.1.6.2 Process Diagram



4.1.6.3 Process Analysis

Determine Requirements

The number of fixtures is determined based on codes and the floor occupancy. The decision on the type of fixtures such as whether the toilet is wall hanging or rests on the floor is also made. A decision on the partition types between the fixture is defined.

Input Information:
- *Occupancy type*
- *Floor area*
- *Municipal fixtures requirements*
- *Special Criteria (list of fixtures)*

Output Information:
- *Fixtures number and types (urinal, WC wall, WC floor etc.)*

Layout

Layout involves the location of the fixtures while creating appropriate circulation for occupants.

Input Information:
- *Fixtures (mounting height, clearances)*
- *accessories (grab bars, mirrors, paper towel, trash, partition etc.) mounting, clearances, width, length, height, depth.*
- *Special Criteria (list of fixtures)*
- *Core constraints (width, length, area, polygonal area)*
- *Structural Grid*

Output Information:
- *Location, height of fixtures and accessories, material*
- *Floor shape/ area*

4.1.6.4 IFC Model Impact

New object types

- *Toilet Partitions*
  - *Height*
  - *Width*
  - *Thickness*
  - *Material*
  - *Finish*
  - *Mounting*
  - *Manufacturer*
- *Lavatories*
  - *Height*
  - *Width*
  - *Thickness*
  - *Material*
  - *Finish*
  - *Mounting*
  - *Manufacturer*
- *Urinals*
  - *MountingHeight*

  - *Width*
  - *Mounting*
  - *Manufacturer*

- *Mirrors*
  - *Height*
  - *Width*
  - *Thickness*
  - *Material*
  - *Mounting*
  - *Manufacturer*

- *Faucet*
  - *MountingLocation*
  - *Material*
  - *Finish*
  - *Manufacturer*

- *Counter*
  - *Height*
  - *Width*
  - *Depth*
  - *Material*
  - *Mounting*

Toilets
Sinks
Accessories
Signage
Dispenser
Towels
Coat Hooks
Grab Bars
Changing Table
Water Fountains
Sinks
Shelves
Benches
Lighting Fixtures

Extensions to R1.0 object types

- *{{ Object type name }}*
  - *{{ attribute 1 }}*
    - *{{ attribute 2 }}*

- *{{ Object type name }}*
- *{{ attribute 1 }}*
- *{{ attribute 2 }}*

4.1.6.5 RoadMap Issues

Interoperability issues

Disciplines from which information is needed:

- *Structural*
- *Plumbing*
- *HVAC*
- *Electrical*

Disciplines for which information is produced:

- *HVAC*
- *Plumbing*
- *Structural*
- *Electrical*
- *Construction*
- *Facility Management*
- *Specifications*

4.2 AR-2 Space Planning for Escape Routes

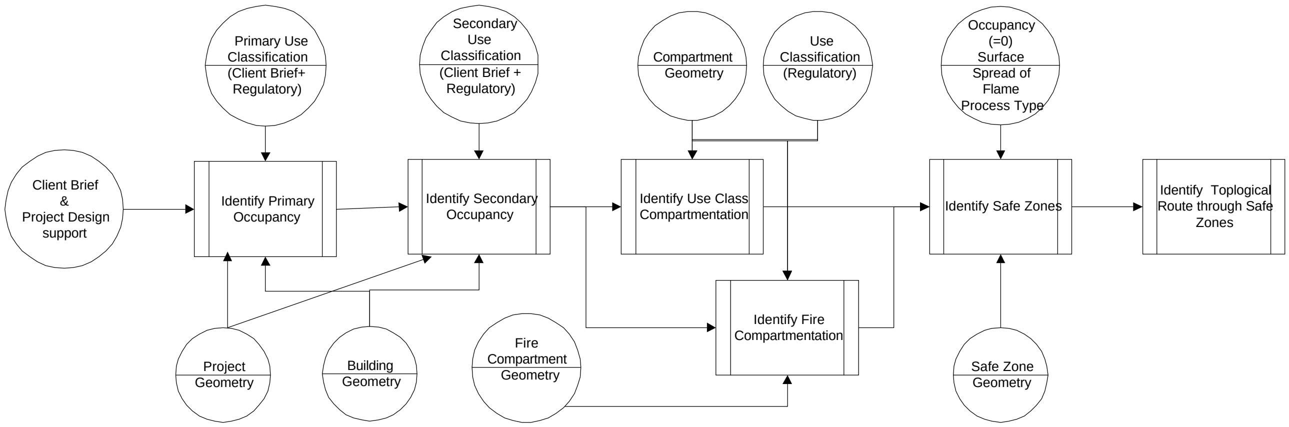
4.2.1 Means of Escape from Spaces

Overview: Means of escape from an occupied space to a safe space. The process will identify occupancy Use class/classes. compartments, and safe zones. Then the process will define space, openings and adjacent spaces, providing an escape route to safe space.

4.2.1.1 Industry Process Definition

On receipt of Architects drawings, first identify primary and secondary Use classes, for the total project. In doing so define shape and size of each Use class. Subject to numbers of Use classes, and their shape & size. define Use class compartments. Then sub-divideUse class compartments into Fire compartments. and Safe (*escape*) spaces. Then extend the Safe escape space routes through Safe spaces, to the final safe space, i.e. external space.

4.2.1.2 Process Diagram



4.2.1.3 Process Analysis

Identify Primary Occupancy

Identify the Primary occupancy of space as dictated by the functional role of that space.

Input Information:

- *Client Brief & Project Design*
    - *Project Usage Type*
    - *Project Geometry*
    - *Building Usage Type*
    - *Building Geometry*
    - *Inter Building relationship*
    - *Inter Usage relationship*
    - *Design Occupancy Number per Building and Usage Type*
    - *Actual Occupancy Number per Building and Usage Type*
    - *Design Occupancy Type or Types per Building and Usage Type*

Output Information:

- *Identify Primary Occupancy*
    - *Primary Usage Type*
    - *Building Geometry*
    - *Building Occupancy Type or Types*
    - *Design Occupancy Number*
    - *Actual Occupancy Number*
    - *Relationship to other Primary Usage Types)*
    - *Relationship to other Secondary Usage Types)*

Project Model Requirements:

Existing classes:

IfcProject

IfcSite

IfcBuildingComplex

New classes:

IfcProjectGeometry

IfcSiteGeometry

IfcBuildingGeometry

IfcProjectUsageType

IfcBuildingUsageType

IfcInterBuildingRelationship

IfcInterUsageRelationship

IfcDesignOccupancyNumber

IfcActualDesignOccupancyNumber

IfcUsageType

IfcPrimaryUsageType

IfcBuildingOccupancyType

IfcPrimaryUsageRelationship

Identify Secondary Occupancy

Identify the Secondary occupancy of space as dictated by the functional role of that space.

Input Information:

- *Client Brief & Project Design*
- *Project Usage Type*
- *Project Geometry*
- *Building Usage Type*
- *Building Geometry*
- *Inter Building relationship*
- *Inter Usage relationship*
- *Design Occupancy Number per Building and Usage Type*
- *Actual Occupancy Number per Building and Usage Type*
- *Design Occupancy Type or Types per Building and Usage Type*

Output Information:

- *Identify Secondary Occupancy*
- *Secondary Usage Type*
- *Building Geometry*
- *Building Occupancy Type or Types*
- *Design Occupancy Number*
- *Actual Occupancy Number*

- *Relationship to the Primary Usage Type)*
- *Relationship to other Secondary Usage Types)*

Project Model Requirements:

Existing classes:

New classes:

IfcSecondaryUsageType

Identify Use Class Compartmentation

Identify the Use Class Compartments by establishing relationships between Primary and its Secondary Usages.

Input Information:

- *Primary and Secondary Occupancies*
- *Primary Usage Type*
- *Secondary Usage Type*
- *Primary Building Geometry*
- *Secondary Building Geometry*
- *Primary Building Occupancy Type or Types*
- *Secondary Building Occupancy Type or Types*
- *Primary and Secondary Design Occupancy Number*
- *Primary and Secondary Actual Occupancy Number*
- *Relationship to the Primary Usage Type)*
- *Relationship to other Secondary Usage Types)*

Output Information:

- *Identify Use Class Compartmentation*
- *Total Design Occupancy number*
- *Total Actual Occupancy number*
- *Occupancy Type or Types*
- *Fittings, Fixtures and local Plant*
- *Services to serve local plant, fittings and fixtures*
- *Compartment Geometry*
- *Compartment Area*
- *Compartment Volume*
- *Compartment Linear Dimensions*

Project Model Requirements:

Existing classes:

New classes:

IfcUseClassCompartmentation

IfcTotalDesignOccupancy

IfcActualDesignOccupancy

IfcFFEP

IfcServicesRelationship

IfcCompartmentGeometry

IfcCompartmentArea

IfcCompartmentVolume

IfcCompartmentLinearDims

Identify Fire Compartmentation

Identify the Fire - Compartments by Use Classifiaction and Regulatory limits

Input Information:

- *Use Class Compartmentation*

Output Information:

- *IdentifyFire Compartments*
    - *Regulatory Fire Compartments*
    - *Relationship between adjecant Fire Compartments*
    - *Relationship between Fire Compartments and other Internal Spaces*
    - *Relationship between Fire Compartments and External Spaces*
    - *Total Fire Compartment Design Occupancy number*
    - *Total Actual Fire Compartment Occupancy number*
    - *Fire Compartment Occupancy Type or Types*
    - *Fittings, Fixtures and local Plant*
    - *Services to serve local plant, fittings and fixtures*
    - *Fire Compartment Geometry*
    - *Fire Compartment Area*
    - *Fire Compartment volume*
    - *Fire Compartment linear dimensions*

Project Model Requirements:

Existing classes:

New classes:

IfcFireRegulations

IfcRegulatoryFireCompartment

IfcTotalRegulatoryFireCompartmentDesignOccupancyNumber

IfcTotalRegulatoryFireCompartmentActualOccupancyNumber

IfcRegulatoryFireCompartmentGeometry

IfcRegulatoryFireCompartmentArea

IfcRegulatoryFireCompartmentVolume

IfcRegulatoryFireCompartmentLinearDims

Identify Safe Zones

Identify the Safe Zones by exaimining areas not taken up by regulatory compartments.

Input Information:

- *Fire Compartmentation*
- *Use Class Compartmentation*

Output Information:

- *IdentifySafe Zones*
  - *Regulatory Internal Safe Spaces within and adjacant to Fire Compartment*
  - *Regulatory External Safe Space adjacant to Fire Compartment*

Project Model Requirements:

Existing classes:

New classes:

IfcInternalSafeSpace

IfcExternalSafeSpace

Identify Topological Route through Safe Zones

Identify the Escape Route by linking Safe Spaces from internal Fire Compartments to the External Safe Spaces.

Input Information:

- *Safe Zones*

Output Information:

- *Identify Escape Route*
    - Link Safe Spaces from interior to the External Safe Spaces

Project Model Requirements:

Existing classes:

-

New classes:

- *IfcSafeSpaceLinks*

4.2.1.4 IFC Model Impact

4.2.1.5 RoadMap Issues

Interoperability Issues

Disciplines from which information is needed:

- *Client Brief*
- *Architecture Building Model)*
- *Services Engineers)*

Disciplines for which information is produced:

- *Architecture*

Target Software Companies/Application Type

- *Architects and Fire Officers*
- *CAD systems providers (e.g. Autodesk)/Autocad*
- *CAD-support FM applications /space planning, occupancy planning, and asset management databases*

Value to AEC Domains

- *Architecture: High (in the top 5)*
- *FM: Very High (in the top 3)*
- *CM/Cost: Very High (in the top 3)*
- *Building Service:*
- *HVAC:*

Sponsor Software Companies

- *None so far*
-

Building Services

4.3 BS-1 HVAC Systems Design

HVAC Duct System Design supports the design and representation of air distribution ductwork systems. These processes are typically performed by engineers during the design phase of a building or project, prior to construction. The process culminates with a set of drawings which can be bid upon and constructed.

4.3.1 HVAC Duct System Design

4.3.1.1 Industry Process Definition

Once an appropriate system type is determined (outside of scope), the HVAC Duct System Design process begins by selecting and locating air terminals, terminal boxes and fans that will be part of the system. An architect will often provide a preliminary reflected ceiling plan that shows desirable locations for air terminals, light fixtures, and sprinkler heads. In the absence of these plans, the HVAC Duct System Designer will select locations for the air terminals using a ceiling or floor plan, and submit these locations to other members of the design team for coordination. To appropriately locate the terminal boxes, a structural plan is required so that initial interferences may be avoided.

The next step in the process is to connect the air terminals, terminal boxes, and fans together with duct and fittings. A graphical representation of this system layout is generated for use in coordination with other disciplines.

The room air flow rates are then assigned to the air terminals. These airflow rates are determined by building load calculations, and these processes are defined in the IFC 1.x Specifications.

The duct and fitting sizes will then be calculated based on these airflow rates and duct system design criteria. The duct and fitting sizes are then updated in the graphical representation of the system.

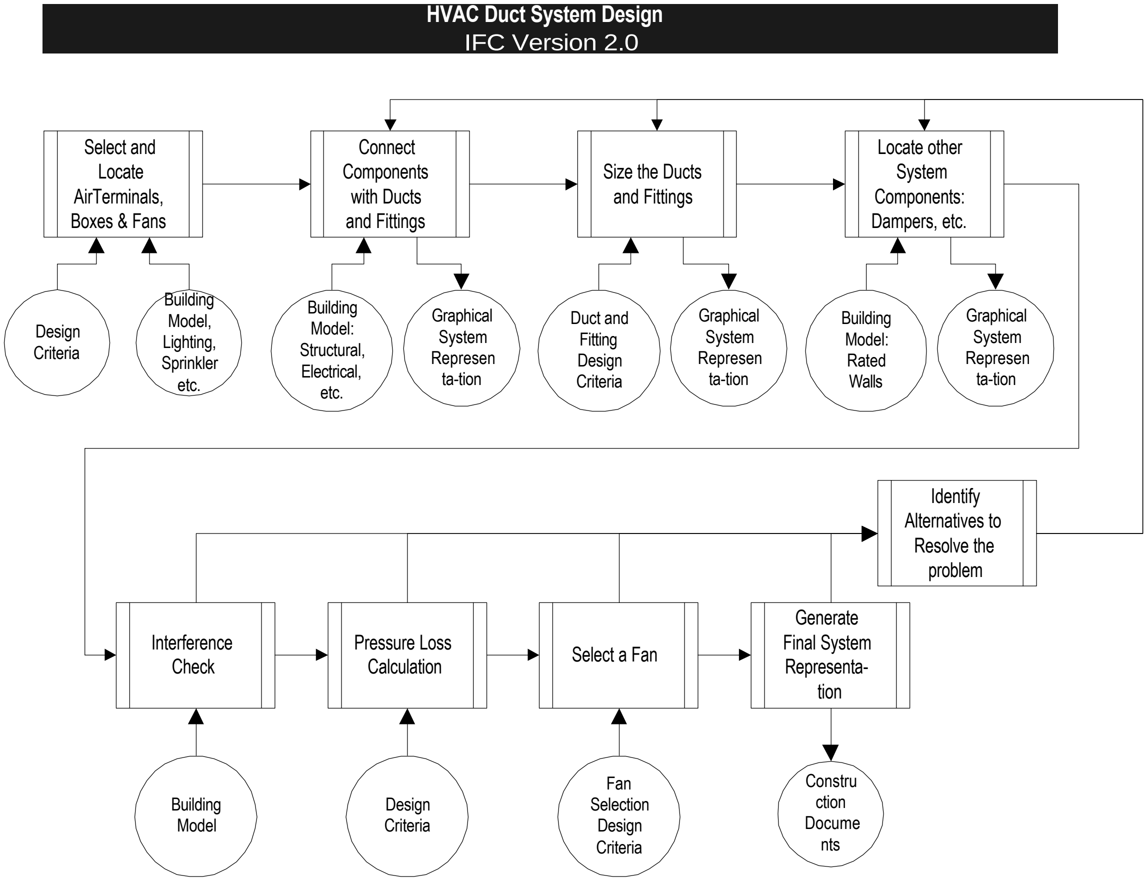
Other required system components, such as dampers, sensors, etc. are then located on the graphical representation. This process requires that fire rated walls, exit corridors, etc. are available from the architectural plans. Any components that require other disciplines to respond are identified, such as electrical power required to motorized dampers.

Once these components are located, an interference check (outside of scope) is performed to identify any locations where the sized duct, fittings, and components may interfere with other design elements. This requires the coordination of all building trades, including electrical, plumbing and fire protection, structural, architectural, etc. If at any point a conflict is observed, a design solution must be identified and implemented, which may require resizing or relocating sections of duct, fittings, etc.

Upon completing this final coordination, the duct system pressure loss calculation is performed (outside of scope). A suitably sized fan may then be selected. This process may also warrant a change to some aspect of the design, depending on the availability of the fan to meet the required performance criteria.

Finally, the coordinated system is represented on a set of contract documents consisting of drawings and specifications. These contract documents are used to bid and construct the system.

4.3.1.2 Process Diagram



4.3.1.3 Process Analysis

Select and Locate Air Terminals, Boxes, and Fans

This step involves selecting and locating the air terminals, boxes (if included in the design), and fans that compose the HVAC duct system.

Input Information:

- *Floor plans*
- *Ceiling grid plans*
- *Reflected ceiling plans*
- *Lighting plans*
- *Structural plans*
- *Sprinkler plans*
- *Smoke detector plans*
- *Speaker plans*

output Information:

- *Att_AirTerminalDevice*
- *Att_CoordinationRequirement*
- *Att_TerminalBox*
- *IfcDevice*
- *IfcEquipment*

Connect the components with Ducts and Fittings

This step involves preparing drawings or specifications which will schematically represent the system under design. These schematics are then used to begin coordination with other disciplines which are impacted by the system.

Input Information:

- *Sprinkler plans*
- *Floor plans*
- *Ceiling grid plans*
- *Reflected ceiling plans*
- *Lighting plans*
- *Structural plans*
- *Att_CoordinationRequirement*

output Information:

- *Att_DuctFitting*
- *Att_DuctSegment*
- *IfcPathwayComponent*
- *IfcPathwayConnector*
- *IfcPathwayObject*
- *IfcPathwayPort*
- *IfcPathwaySegment*

Size the Duct and Fittings (Not in Scope)

The sizes of the duct and fittings are calculated.

Input Information:

- *HVAC Room Load Calculation Information*
- *Att_DuctSystemDesignCriteria*
- *Att_DuctDesignCriteria*

output Information:

- *Att_CoordinationRequirement*

Locate Other System Components

Identify and locate other system components required for the duct system.

Input Information:

- *Att_DuctSystemDesignCriteria*
- *Att_DuctDesignCriteria*

output Information:

- *Att_Actuator*
- *Att_CoordinationRequirement*
- *Att_Damper*
- *Att_Sensor*
- *IfcDevice*
- *IfcPathwayComponent*
- *IfcPathwayObject*
- *IfcPathwayPort*

Interference Check (Not in Scope)

Identify any interferences with other trades.

Input Information:
- *Plumbing/Sprinkler plans*
- *Floor plans*
- *Ceiling grid plans*
- *Reflected ceiling plans*
- *Lighting plans*
- *Power plans*
- *Structural plans*
- *Att_CoordinationRequirement*

output Information:
- *Att_CoordinationRequirement*

Identify alternatives to resolve the problem (Not in Scope)

This step requires the designer to go back and redesign certain portions of the system. This may involve regenerating the schematic design documents and recalculating system component sizes. Note that this step may occur at any point in the process.

Input Information:
- *Att_CoordinationRequirement*

output Information:
- *Att_CoordinationRequirement*

Pressure Loss Calculations (Not in Scope)

Determine the system pressure losses based on the duct system that has been designed.

Input Information:
- *Att_DuctSystemDesignCriteria*
- *Att_DuctDesignCriteria*

output Information:
- *Att_CoordinationRequirement*

Fan Selection (Not in Scope)

Identify a fan that will appropriately meet the requirements of the duct system.

Input Information:
- *Att_DuctSystemDesignCriteria*
- *Att_HVACAirSideSystemDesignCriteria*

output Information:
- *Att_CoordinationRequirement*

Generate Final System Representation

This step involves preparing drawings or specifications which will be used as contract documents for bid and construction. These documents complete the design phase of the system.

Input Information:

- *Att_CoordinationRequirement*

output Information:

- *Att_AirTerminalDevice*
- *Att_TerminalBox*
- *Att_Damper*
- *Att_DuctFitting*
- *Att_DuctSegment*
- *Att_Sensor*
- *IfcDevice*
- *IfcPathwayComponent*
- *IfcPathwayConnector*
- *IfcPathwayObject*
- *IfcPathwayPort*
- *IfcPathwaySegment*

4.3.1.4 IFC Model Impact

Project Model Usage Requirements:

Existing Classes and Attribute Sets:

- ***Att_AirHandlingUnit}***
  Data
  - *To be determined*
  Behavior
  - *To be determined*
- ***Att_Fan***
  Data
  - *To be determined*
  Behavior
  - *To be determined*
- ***IfcConnector***
  Data
  - *To be determined*
  Behavior
  - *To be determined*
- ***IfcConnectionPorts***
  Data
  - *need to reconcile attributes and behavior with IfcPathwayPort*
  Behavior
  - *To be determined*
- ***IfcEquipment***
  Data
  - *need to reconcile relationship with IfcPathwayObject to allow for connectivity*
  Behavior
  - *To be determined*
- ***Att_AirSideSystemInformation***

Data

- *need to coordinate with duct system design*

Behavior

- *To be determined*

- ***Att_Insulation***

Data

- *need to coordinate with duct and pipe insulation*

Behavior

- *To be determined*

New Core Classes and Attribute Sets:

- ***IfcDevice***

This class supports devices and is a subtype of IfcPathwayObject. A device is typically considered a terminus within the system, and in addition to having relationships with its interconnected system components (defined in the supertype), it has a mounting relationship with other building elements such as a wall or ceiling. This relationship allows the device to appropriately move if the object it is 'mounted' upon is moved, while maintaining its system interconnectivity. This class provides a reference to a device type definition which contains the attributes required for the system being designed.

Data

Attribute	Description	Data Type
DeviceType	Named type of Device --> keys to a Device TypeDef which links to attributes shared by all instances of this type.	Ref[IfcTypeDefinition]
IsMountedOn	IfcObject that the device is mounted upon or attached to.	Ref[IfcObject]
Others??	Need to migrate common attribute set attributes to this class...	

Behavior

- *To be determined*

- ***IfcPathwayComponent***

This class supports components other than pathway segments, pathway connectors, equipment, or devices that are integrated into the system. It is a subtype of IfcPathwayObject. An example of a pathway component is a flow controller, such as a valve or damper. This class provides a reference to a pathway component type definition which contains the attributes required for the system being designed.

Data

Attribute	Description	Data Type
PathwayComponentType	Named type of PathwayComponent --> keys to a PathwayComponent TypeDef which links to attributes shared by all instances of this type.	Ref[IfcTypeDefinition]
Others??	Need to migrate common attribute set attributes to this class...	

Behavior

- *To be determined*

- ***IfcPathwayConnector***

This class supports pathway connectors such as tees and elbows, and is a subtype of IfcPathwayObject. The pathway connector connects other components in the system, such as pathway segments, pathway components, equipment, or devices This class provides a reference to a pathway connector type definition which contains the attributes required for the system being designed.

Data

Attribute	Description	Data Type
PathwayConnectorType	Named type of PathwayConnector --> keys to a PathwayConnector TypeDef which links to attributes shared by all instances of this type.	Ref[IfcTypeDefinition]

Others??	Need to migrate common attribute set attributes to this class...	

Behavior

- *To be determined*

- ***IfcPathwayObject***

This is the base class for networked system classes and is derived from IfcManufacturedElement. It contains the basic connectivity references, both physical and logical, that are used to traverse the system network in the direction of flow.

Data

Attribute	Description	Data Type
InletPort	The primary physical inlet port of the system object	Ref[IfcPathway Port]
OutletPort	The primary physical outlet port of the system object	Ref[IfcPathway Port]
BranchPorts	Any branch ports that are a part of the pathway object	List[0:?] of Ref[IfcPathway Port]
Others??		

Behavior

- *To be determined*

- ***IfcPathwayPort***

This class contains the information about a port in a system object. A port is an object that is used to interconnect IfcPathwayObjects.

Data

Attribute	Description	Data Type
Shape	Physical shape of the port (i.e., round, rectangular, oval, etc.)	IfcString
NominalSize	Nominal size of the port	IfcSize
PhysicalSize	Physical size of the port; may be the same as NominalSize	IfcSize
Type	Physical type of the port (i.e., flanged, screwed, welded, etc.)	IfcString
PhysicalLocation	Physical location of the port	IfcOrientedVert ex
LogicalLocation	Logical location of the port	IfcOrientedVert ex
IsConnectedTo	List of references the port is connected to	List[0:?] of Ref[IfcPathway Object]
Others??		

Behavior

- *To be determined*

- ***IfcPathwaySegment***

This class supports pathway segments such as lengths of pipe or duct, and is a subtype of IfcPathwayObject. This class provides a reference to a pathway segment type definition which contains the attributes required for the system being designed.

Data

Attribute	Description	Data Type
PathwaySegmentType	Named type of PathwaySegment --> keys to a PathwaySegment TypeDef which links to attributes shared by all instances of this type.	Ref[IfcTypeDefi nition]
Others??	Need to migrate common attribute set attributes to this class...	

Behavior

- *To be determined*

- ***Att_CoordinationRequirement***

This attribute set provides a placeholder for interoperable coordination requirements between different disciplines.

Data

Attribute	Description	Data Type
OriginatingDiscipline	The discipline which originates the coordination requirement	IfcString
AffectedDiscipline	The discipline which must act upon the coordination requirement	IfcString
Requirement	The coordination requirement	IfcString
Others??		

Behavior

- To be determined

New Extension Classes and Attribute Sets:

- ***Att_AirTerminalDevice***

This attribute set will be used by an IfcDevice object.

Data

Attribute	Description	Data Type
Flowrate	Required air flowrate for the terminal device	IfcFlowrate
PressureLoss	Pressure loss through the terminal device	IfcPressure
AirDirection	Direction of airflow into the terminal device: in, out, Supply, Return, Exhaust, etc.	IfcString
MountingFrame	Frame for plaster, drywall, lay-in grid, etc.	IfcString
AdjustableCore	Permits adjustment of throw	IfcString
CoreSetHorizontal	Degree of blade set from the centerline	IfcAngle
CoreSetVertical	Degree of blade set from the centerline	IfcAngle
IntegralDamper	Volume damper combined with terminal device	IfcBool
IntegralControl	Self powered temperature control	IfcBool
SoundLevel	Design sound power level	IfcString

Behavior

- To be determined

- ***Att_Damper***

This attribute set will be used by an IfcPathwayComponent object.

Data

Attribute	Description	Data Type
WorkingPressure		IfcPressure
PressureDrop		IfcPressure
CloseOffRating		IfcPressure
LeakageAirFlowrate		IfcFlowrate

Behavior

- To be determined

- ***Att_TerminalBoxr***

This attribute set will be used by an IfcPathwayComponent object.

Data

Attribute	Description	Data Type
TerminalBoxType	Type of terminal box: VAV, CV, VVRH, etc.	IfcString
Flowrate	Required air flowrate for the terminal box	IfcFlowrate
PressureLoss	Pressure loss through the terminal box	IfcPressure
SoundLevel	Design sound power level	IfcString

Behavior

- To be determined

- ***Att_DuctFitting***

This attribute set will be used by an IfcPathwayConnector object.

Data

Attribute	Description	Data Type
Type	Major type of fitting (i.e., elbow, tee, cross, etc.)	IfcString
SubType	Subtype of fitting (i.e., 5-gore, pleated, stamped, etc.)	IfcString
EnteringPressure	Actual pressure required for balancing and maintenance	IfcPressure

LiningType	Duct lining type if different from SystemDesignCriteria	IfcString
InsulationType	Duct insulation type if different from SystemDesignCriteria	IfcString

Behavior

- *To be determined*

- ***Att_DuctSegment***

This attribute set will be used by an IfcPathwaySegment object.

Data

Attribute	Description	Data Type
Flowrate	Flowrate through the duct	IfcFlowrate
PressureClass	Nominal pressure rating	IfcPressure
LeakageClass	Nominal leakage rating	IfcPressure
AmbientDryBulb	The ambient dry bulb temperature of the space adjacent to the pathway segment	IfcTemperature
Lining	Yes or No, type and thickness are design criteria	IfcString
Insulation	Yes or No, type and thickness are design criteria	IfcString
SizingMethod	If different from SystemDesignCriteria	IfcString
SupportMethod	Hanger or other structural support from roof, floor, etc.	IfcString

Behavior

- *To be determined*

- ***Att_DuctDesignCriteria:***

This attribute set will typically be used in conjunction with Att_Fluid and Att_Insulation.

Data

Attribute	Description	Data Type
DesignName	A name for the design values	IfcString
AspectRatio	The default aspect ratio	IfcReal
SegmentLiningType	The segment lining type	Ref[Att_Insulation]
SegmentInsulationType	The segment insulation type	Ref[Att_Insulation]
ConnectorLiningType	The connector or fitting lining type	Ref[Att_Insulation]
ConnectorInsulationType	The connector or fitting insulation type	Ref[Att_Insulation]
PlenumLoss	The pressure loss in the plenum	IfcPressure
OtherLoss	Other non-specific losses	IfcPressure
ScrapFactor	Sheet metal scrap factor	IfcReal
MainVelocity	The maximum velocity of the air in the main branch	IfcVelocity
BranchVelocity	The maximum velocity of the air in the branch	IfcVelocity
FrictionLoss	The pressure loss due to friction per unit length	IfcPressure/IfcLength
MinimumHeight	The minimum duct height	IfcLength
MinimumWidth	The minimum duct width	IfcLength

Behavior

- *To be determined*

- ***Att_DuctSystemDesignCriteria:***

This attribute set will typically be used in conjunction with Att_Fluid and Att_Insulation.

Data

Attribute	Description	Data Type
SystemType	The type of the system (i.e., VAV, Constant Volume, Double-Duct, etc.)	IfcString
SystemId	Air Handling Unit Identifier	IfcString
SystemLocation	Physical description of the part of the building the system serves	IfcString
SizingMethod	The methodology to be used to size system components.	IfcString

Behavior

- *To be determined*

4.3.1.5 RoadMap Issues

Interoperability Issues

Disciplines from which information is needed:

- *Architectural*
- *Structural*
- *HVAC*
- *Plumbing/Fire Protection*
- *Electrical*
- *Lighting*

Disciplines for which information is produced:

- *Electrical*
- *HVAC*
- *Plumbing/Fire Protection*

Value to AEC Domains

- *Architecture (7)*
- *Building Services (8)*
    - *HVAC (9)*
- *FM (6)*
- *CM/Cost (8)*

Sponsor Software Companies

- *APEC*
- *Carrier*
- *Greenheck*
- *Landis-Staefa*
- *Honeywell*
- *Johnson Controls*

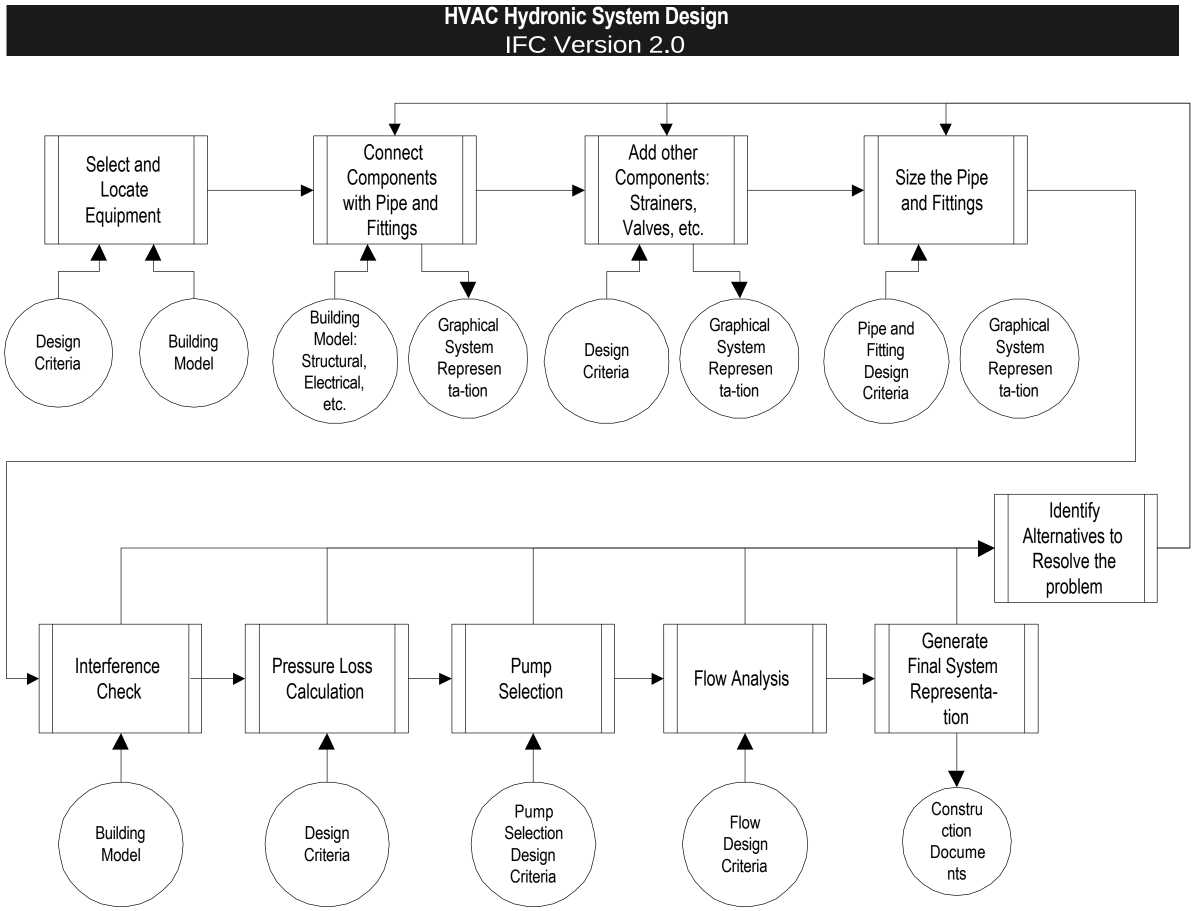
4.3.2 HVAC Hydronic System Design

HVAC Hydronic System Design supports the design and representation of piping systems. These processes are typically performed by engineers and design-build contractors during the design phase of a building or project, prior to construction. The process culminates with a set of drawings which can be bid upon and constructed.

This section defines the specific requirements for HVAC Hydronic System Design based on the generalized Building Services System Design described above.

4.3.2.1 Industry Process Definition

4.3.2.2 Process Diagram



4.3.2.3 Process Analysis

Select and Locate Equipment

This step involves selecting and locating the equipment that composes the HVAC hydronic system.

Input Information:

- *Floor plans*
- *Reflected ceiling plans*
- *Structural plans*
- *Att_PipeSystemDesignCriteria*

output Information:

- *Att_Boiler*
- *Att_Chiller*
- *Att_Coil*
- *Att_CoolingTower*
- *Att_Pump*
- *Att_TerminalBox*
- *Att_TubeBundle*

Connect the components with Pipe and Fittings

This step involves preparing drawings or specifications which will schematically represent the system under design. These schematics are then used to begin coordination with other disciplines which are impacted by the system.

Input Information:

- *plans*
- *Reflected ceiling plans*
- *Floor plans*
- *Ceiling grid plans*
- *Reflected ceiling plans*
- *Lighting plans*
- *Structural plans*
- *Att_CoordinationRequirement*

output Information:

- *Att_PipeFitting*
- *Att_PipeSegment*
- *IfcPathwayComponent*
- *IfcPathwayConnector*
- *IfcPathwayObject*
- *IfcPathwayPort*
- *IfcPathwaySegment*

Locate Other System Components

Identify and locate other system components required for the hydronic system.

Input Information:

- *Att_PipeSystemDesignCriteria*
- *Att_PipeDesignCriteria*

output Information:

- *Att_Actuator*
- *Att_CoordinationRequirement*
- *Att_Valve*
- *Att_Sensor*
- *IfcDevice*
- *IfcPathwayComponent*
- *IfcPathwayObject*
- *IfcPathwayPort*

Size the Pipe and Fittings (Not in Scope)

The sizes of the pipe and fittings are calculated.

Input Information:

- *Att_PipeSystemDesignCriteria*
- *Att_PipeDesignCriteria*

output Information:

- *Att_CoordinationRequirement*

Interference Check (Not in Scope)

Identify any interferences with other trades.

Input Information:
- *Electrical plans*
- *Plumbing/Sprinkler plans*
- *Floor plans*
- *Ceiling grid plans*
- *Reflected ceiling plans*
- *Lighting plans*
- *Structural plans*
- *Att_CoordinationRequirement*

output Information:
- *Att_CoordinationRequirement*

Identify alternatives to resolve the problem (Not in Scope)

This step requires the designer to go back and redesign certain portions of the system. This may involve regenerating the schematic design documents and recalculating system component sizes. Note that this step may occur at any point in the process.

Input Information:
- *Att_CoordinationRequirement*

output Information:
- *Att_CoordinationRequirement*

Pressure Loss Calculations (Not in Scope)

Determine the system pressure losses based on the hydronic system that has been designed.

Input Information:
- *Att_PipeSystemDesignCriteria*
- *Att_PipeDesignCriteria*

output Information:
- *Att_CoordinationRequirement*

Pump Selection (Not in Scope)

Identify a pump that will appropriately meet the requirements of the hydronic system.

Input Information:
- *Att_PipeSystemDesignCriteria*

output Information:
- *Att_CoordinationRequirement*

Flow Analysis (Not in Scope)

Determine whether the hydronic system flows resulting from loads, pressure drops and pump selections will perform as desired.

Input Information:

output Information:
- *Att_CoordinationRequirement*

Generate Final System Representation

This step involves preparing drawings or specifications which will be used as contract documents for bid and construction. These documents complete the design phase of the system.

Input Information:
- *Att_CoordinationRequirement*

output Information:
- *Att_Actuator*
- *Att_TerminalBox*
- *Att_Valve*
- *Att_PipeFitting*
- *Att_PipeSegment*
- *Att_Sensor*
- *IfcDevice*
- *IfcPathwayComponent*
- *IfcPathwayConnector*
- *IfcPathwayObject*
- *IfcPathwayPort*
- *IfcPathwaySegment*

4.3.2.4 IFC Model Impact

Project Model Usage Requirements:

Existing Classes and Attribute Sets:
Refer also to the Duct Design section for common Existing Classes and Attribute Sets

- ***Att_Boiler***
  Data
  - *To be determined*
  Behavior
  - *To be determined*
- ***Att_Chiller***
  Data
  - *To be determined*
  Behavior
  - *To be determined*
- ***Att_Coil***
  Data
  - *To be determined*
  Behavior
  - *To be determined*
- ***Att_CoolingTower***
  Data
  - *To be determined*

Behavior

- To be determined

- ***Att_Pump***

Data

- To be determined

Behavior

- To be determined

- ***Att_TerminalBox***

Data

- To be determined

Behavior

- To be determined

- ***Att_TubeBundle***

Data

- To be determined

Behavior

- To be determined

New Core Classes and Attribute Sets:

Refer to the Duct Design section for New Core Classes and Attribute Sets

New Extension Classes and Attribute Sets:

- ***Att_Valve***

This attribute set will be used by an IfcPathwayComponent object.

Data

Attribute	Description	Data Type
WorkingPressure		IfcPressure
PressureDrop		IfcPressure
CloseOffRating		IfcPressure
ValveCV		IfcReal

Behavior

- To be determined

- ***Att_PipeFitting***

This attribute set will be used by an IfcPathwayConnector object.

Data

Attribute	Description	Data Type
Type	Major type of fitting (i.e., elbow, tee, cross, etc.)	IfcString
SubType	Subtype of fitting (i.e., long-radius, short-radius, etc.)	IfcString
Class	Pressure class (i.e., Schedule 40, 80, Type L, etc.)	IfcString
EnteringPressure	Actual pressure required for balancing and maintenance	IfcPressure
InsulationType	Pipe insulation type if different from SystemDesignCriteria	IfcString
InsulationJacketType	Pipe insulation jacket type if different from SystemDesignCriteria	IfcString

Behavior

- To be determined

- ***Att_PipeSegment***

This attribute set will be used by an IfcPathwaySegment object.

Data

Attribute	Description	Data Type
Flowrate	Flowrate through the pipe	IfcFlowrate
PressureClass	Nominal pressure rating	IfcPressure
AmbientDryBulb	The ambient dry bulb temperature of the space adjacent to the pathway	IfcTemperature

	segment	
Insulation	Yes or No, type and thickness are design criteria	IfcString
SizingMethod	If different from SystemDesignCriteria	IfcString
SupportMethod	Hanger or other structural support from roof, floor, etc.	IfcString

Behavior

- To be determined

- ***Att_PipeSystemDesignCriteria***

This attribute set will typically be used in conjunction with Att_Fluid and Att_Insulation.

Data

Attribute	Description	Data Type
SystemType	The name of the system (i.e., cooling water, domestic hot water, etc.)	IfcString
SystemId	Riser number, Pump, fan, AHU ID, other -- ???	IfcString
SystemLocation	Physical description of the part of the building the system serves	IfcString
FluidSourcePressure	Pressure in main for domestic water, sprinklers, system pressure for hydronic systems, etc.	IfcPressure
FluidDesignPressure	Steam, hot water, gas, etc.	IfcPressure
FluidLiftHeight	Lift that may be required on open systems with dense fluid.	IfcLength
SizingMethod	The default methodology to be used to size system components.	IfcString

Behavior

- To be determined

- ***Att_PipeDesignCriteria***

This attribute set will typically be used in conjunction with Att_Fluid and Att_Insulation.

Data

Attribute	Description	Data Type
DesignName	A name for the design values	IfcString
SizingMethod	The sizing method to be used if different from the system design criteria	IfcString
MaximumVelocity	The maximum allowable fluid velocity	IfcVelocity
SegmentInsulationType	The segment insulation type	Ref[Att_Insulation]
ConnectorInsulationType	The fitting or connector insulation type	Ref[Att_Insulation]
OtherLoss	Other non-specific losses	IfcPressure

Behavior

- To be determined

4.3.2.5 RoadMap Issues

Integration issues

Disciplines from which information is needed:

- *Architectural*
- *Structural*
- *HVAC*
- *Plumbing/Fire Protection*
- *Electrical*

Disciplines for which information is produced:

- *Electrical*
- *HVAC*
- *Plumbing/Fire Protection*

Value to AEC Domains

- *Architecture (7)*

- *Building Services (8)*
    - *HVAC (9)*
- *FM (6)*
- *CM/Cost (8)*

Sponsor Software Companies

- *APEC*
- *Carrier*
- *Greenheck*
- *Landis-Staefa*
- *Honeywell*

4.4 Power and Lighting Systems Design

{{ Model Requirements Analysis for this project not yet available }}

4.5 BS-3 Pathway Design and Coordination

{{ Model Requirements Analysis for this project not yet available }}

4.6 BS-4 HVAC Loads Calculation

{{ Model Requirements Analysis for this project not yet available }}

Client Briefing

4.7 CB-1 Client Briefing

{{ This project has been delayed for inclusion in IFC Release 3.0 }}

Construction / Construction Management

4.8 CM-1 Procurement and Logistics

{{ This project has been delayed for inclusion in IFC Release 3.0 }}

4.9 CM-2 Temporary Construction

{{ This project has been delayed for inclusion in IFC Release 3.0 }}

Codes and Standards

4.10 CS-1 - Code Compliance Enabling Mechanism/Energy Code Compliance Checking

4.10.1 Commercial and Residential Energy Code Compliance Checking

Code compliance is performed by building designers, systems designers, and code enforcement officials. Compliance with codes begins during programming when designers determine which codes apply to the building project. Preliminary code reviews are frequently performed during schematic design and more thorough reviews are performed by members of the design team late in the design process before construction documents are complete. Building code officials perform plan reviews as part of the building permit process. Designers and code official perform drawing takeoffs as necessary to ensure compliance. Information about building systems, assemblies, layout, etc. is gathered during this process and compared to the requirements for each applicable code. Virtually all systems within a building are constrained in some way by codes (or voluntary design standards), hence codes are relevant to most other design processes. Energy codes, the subject of this Release 2.0 proposal, are strongly related to architectural, HVAC, and electrical design processes.

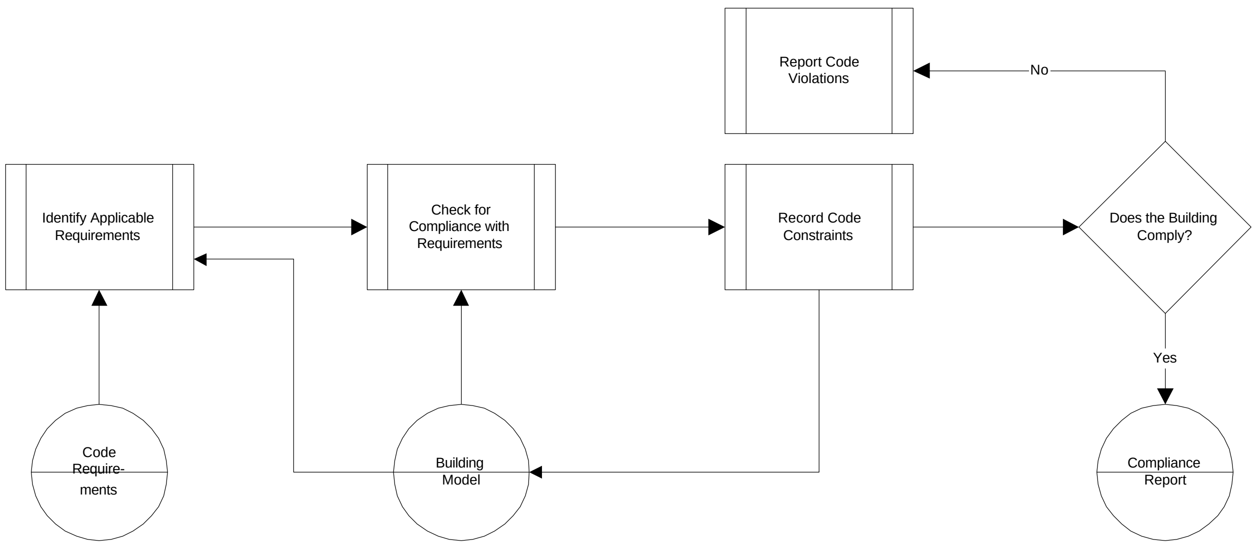
4.10.1.1 Industry Process Definition

Code compliance is the process of assessing whether a building complies with codes enforced by local jurisdictions or with voluntary design standards promulgating by various standard-writing entities.

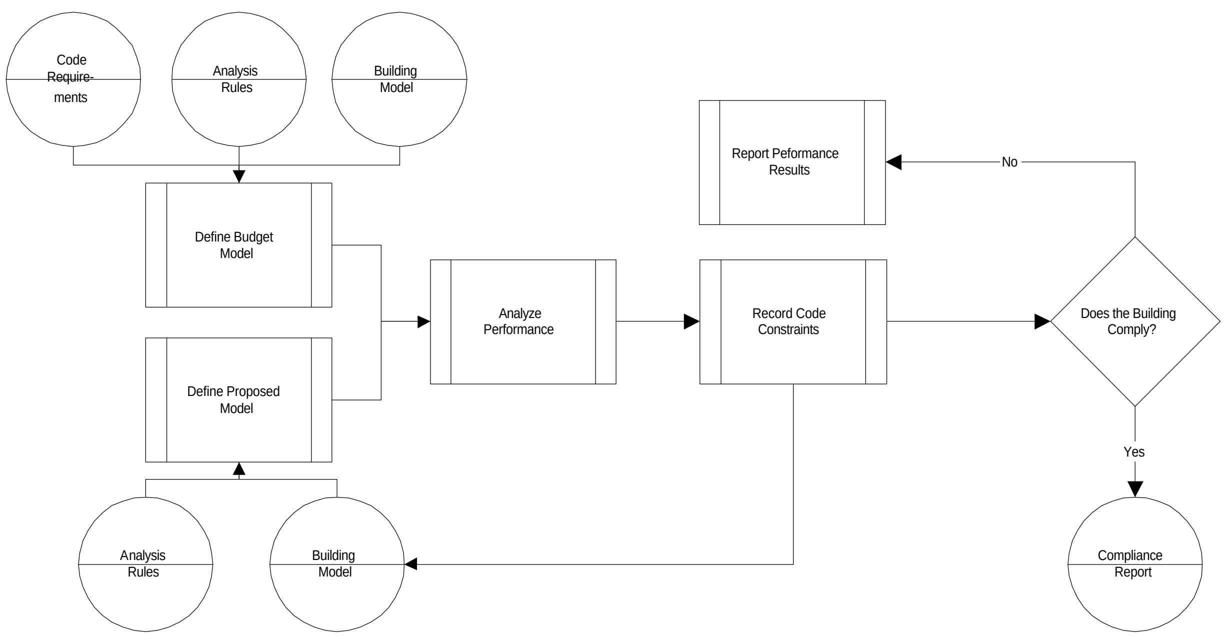
4.10.1.2 Process Diagram

Note: Building codes typically employ two different regulatory approaches: prescriptive requirements and performance requirements. The two-part diagram below illustrates the two different processes corresponding with these two different approaches. Most codes are neither purely prescriptive nor purely performance-based, but rather, contain elements of both types of requirements either in combination or as alternative paths for demonstrating compliance.

Process for Prescriptive Code Requirements



Process for Performance Code Requirements



4.10.1.3 Process Analysis

The processes illustrated above will be employed in code checking applications that address the following codes:

1. ASHRAE/IESNA Standard 90.1-1989 (Std 90.1)
2. Model Energy Code (MEC - all recent years)

The specific tasks illustrated in the diagrams above are all embedded within existing widely-distributed applications.

4.10.1.3.1 Commercial and Residential Energy Code Compliance Checking

The inputs and outputs of the individual process tasks are not generally shared with other applications and are too numerous to be conveniently listed as separate task inputs and outputs using this format. For now, they have simply be summarized for the entire process below. Use of existing classes has not been noted, except where new attributes are required. Because the product model usage requirements are not broken down by task, they are identical to IFC Model Impact section, and the information is shown once there. [This information is also shown in an accompanying spreadsheet table.]

Input Information:

- *Code Requirements*
- *Building Model*
- *Analysis Rules*

output Information:

- *Object Constraints*
  - *IfcPropetyConstaints*
  - *IfcIntent*
      - *IfcAgregateControl*
- *Code Violation Reports*
- *Compliance Performance Results*
- *Compliance Reports*

4.10.1.4 IFC Model Impact

This section summarizes the model requirements from all the process tasks in two groups à Extensions to R1.0 model object types and proposed new object types for R2.0.

Usage/Extensions to R1.0 object types

- ***IfcLayeredElement (Interfaces added to core class)***

  Data

  - *ContinuousRvalue - Ifcreal*
    - *Continuous R-value of assembly including air films, cladding, gypsum board and sheathing layers*
  - *AssemblyUfactor - Ifcreal*
    - *Overall assembly U-factor*

- ***IfcMaterialLayerSet (Interfaces added to resource schema)***

  Data

  - *ContinuousRvalue - Ifcreal*
    - *Continuous R-value of assembly including air films, cladding, gypsum board and sheathing layers*

- *AssemblyUfactor - Ifcreal*
  - *Overall assembly U-factor*
- *ParallelLayer1Material - Ref[IfcMaterialLayer]*
  - *Reference to first part of parallel portion of layered assembly*
- *ParallelLayer2Materiall - Ref[IfcMaterialLayer]*
  - *Reference to second part of parallel portion of layered assembly*
- *AspectRatioOfLayers - Ifcreal*
  - *Ratio of layer 1 to layer 2*

- ***IfcMaterialLayer (Interfaces added to resource schema)***

  Data
  - *MaterialType - Ref[IfcMaterialTypeLibraryEntry]*
    - *Type reference for homogenous material -- only used if not a material set*
  - *MaterialSet - Ref[IfcMaterialLayerSet]*
    - *Set of materials for material -- only used if not a homogenous material*

- ***IfcWall (Attribute added to core class)***

  Data
  - *AboveGrade - IfcReal*
    - *Ratio of wall area that is above grade to total wall area*

- ***IfcRoof (This class to replace IfcRoofSlab because there are several other roof types)***

  Data
  - *GenericType - IfcRoofTypeEnum*
    - *Predefined generic types are specified in an Enum. A Type definition is available for each generic type (as the required attributes differ). Use TypeDefinition corresponding to this generic type.*
  - *RoofType - Ref[IfcfTypeDefinition]*
    - *Reference to a type definition that links to attributes defining the element (either shared by all instances or added to the ExAttributeSets). Specific TypeDef determined by the Generic Type above.*

- ***IfcFillingElement***

  Data
  - *FillingElementType - [Ref [IfcFillingElementTypeLibraryEntry]*
    - *Predefined generic filling element types specified in a library*
  - *ProjectionFactor - IfcReal*
    - *The ratio of shading projection depth to the height of window*

New object types required

- ***IfcPropertyConstraint (To establish a specific limit on an object or attribute of an object)***

  Data
  - *Source - IfcOwnerId*
    - *Code/Standard reference*
  - *ReferenceObject - IfcProjectObject/ IfcAttributeObject*
    - *Object / attribute reference for which the constraint is specified*
  - *Relation - IfcNumericRelation*

- *ConstraintType - IfcConstraintLevel*
- *NoticeText - IfcString*

• ***IfcIntent (A collection of attributes representing design intent)***

Data

- *Source - IfcOwnerId*
  - *Code/Standard reference ??*
- *Description - IfcString*
  - *Description of the code requirement*

• ***IfcAggregateControl (A collection of attributes representing the logical relationships between design intent and constraint)***

Data

- *Source - IfcOwnerId*
  - *Code/Standard reference*
- *Operation - IfcLogicalOperation*
  - *Logical relationship between intent and constraint*

• ***IfcBuildingEnvelope***

Data

- *AggregateOf - Set[0:N] Ref[IfcLayeredElement]*
  - *Contains references to all instances of layered elements which form the envelope*
- *OccupancyType - IfcEnvelopeOccupancyTypeEnum*
  - *Envelope occupancy type according to the Standard*
- *InternalLoadDensity - IfcReal*
  - *Total internal load based on the occupancy*
- *ThermalLoad - IfcReal*
  - *Envelope load based on the proposed design*

• ***IfcSkylight***

Data

- *GenericType - IfcSkylightTypeEnum*
  - *Predefined generic types are specified in an Enum. A Type definition is available for each generic type (as the required attributes differ). Use TypeDefinition corresponding to this generic type.*
- *SkylightType - Ref[IfcTypeDefinition]*
  - *Reference to a type definition which links to attributes defining the element (either shared by all instances or added to the ExAttributeSets). Specific TypeDef determined by the GenericType above.*

• ***IfcLightingElement (An aggregation class containing all the lighting fixtures)***

Data

- *ReferenceObjects - Ref[IfcFixture]*
  - *Contains references to all instances of IfcFixture that are part of the lighting system*
- *OccupancyType - IfcLightingOccupancyType*
  - *Lighting occupancy type according to the Standard*
- *LightingPowerDensity - IfcReal*
  - *Lighting power density specified by the Code (based on Occupancy type)*

- *LighingPower - IfcReal*
  - *Total lighting power for the proposed design*

- ***IfcLightingFixture***

  Data
  - *Category - Ref[IfcLightingFixtureType LibraryEntry]*
    - *The category of lighting fixture*
  - *NumberOfLampsPerFixture - IfcReal*
    - *Number of lamps per fixture*
  - *FixtureIdentification - IfcString*
    - *Fixture identification on plan*
  - *FixtureWattage - IfcInteger*
    - *Total input wattage of the fixture including lamps and ballast*
  - *NumberOfFixtures - IfcInteger*
    - *Total number of this fixture type used in the building*

- ***IfcMaterial Type (Class structure for material properties library--Not addition to Core class)***

  Data
  - *Type - IfcMaterialTypeEnum*
    - *Describes the function of the material layer as an Enum*
  - *ThermalResistance - IfcReal*
    - *Thermal resistance of the material for unit thickness*
  - *HeatCapacity - IfcReal*
    - *Specific heat capacity of the wall material*

- ***IfcFillingElementType (Class structure for filling element library--Not addition to Core class)***

  Data
  - *FramingType - IfcFrameTypeEnum*
    - *Enum representing the frame type*
  - *GlazingType - IfcGlazingTypeEnum*
    - *Enum representing the glazing type*
  - *ThermalResistance - IfcReal*
    - *Thermal resistance of the filling material*
  - *ShadingCoefficient - IfcReal*
    - *Shading coefficient of filling material*

- ***IfcLightingFixtureType (Class structure for lighting fixture library--Not addition to Core class)***

  Data
  - *Description - IfcString*
    - *Description of the lighting Fixture*
  - *LampType - IfcLampTypeEnum*
    - *Lamp type*
  - *LampDescription - IfcString*
    - *Description of the lamp type*
  - *WattagePerLamp - IfcInteger*

  - *Power used by each lamp in the fixture*
- *BallastType - IfcBallastTypeEnum*
  - *The type of ballast used in the fixture*

4.10.1.5 RoadMap Issues

Interoperability Issues

Applications from which information is needed:

- *Architecture*
- *HVAC*
- *Lighting*

Applications for which information is produced:

- *Architecture*
- *HVAC*

Lighting

Value of software supporting this process

{{In this section, please allow for the other domains to rank your process in order of precedence for their domain, this allows us to examine the issue on a group as well as an individual level}}

- *{{ discipline 1 }} - {{value from 1-10, 1 being the highest value, 10 being the lowest]]*
- *{{ discipline 2 }} - {{value from 1-10]]*

Sponsor Software Companies

Software Companies that have shown an interest in developing applications which implement the process

- *Pacific Northwest National Laboratory*
- *Softdesk*

4.10.1.6 Issues identified in reviews

{{ Reviewing group - Reviewed for: }}

- ***Issues:***
  - ***{{ Proposed resolution }}***
- **{{ Issue 1 }}**
  - *{{ Proposed resolution }}*
- **{{ Issue 2 }}**
  - *{{ Proposed resolution }}*

4.11 CS-2 Code Checking Extensions

4.11.1 Code Extensions (Disabled Access and Escape Routes)

(Code Compliance - Disabled Access & Escape Routes)

4.11.1.1 Industry Process Definition

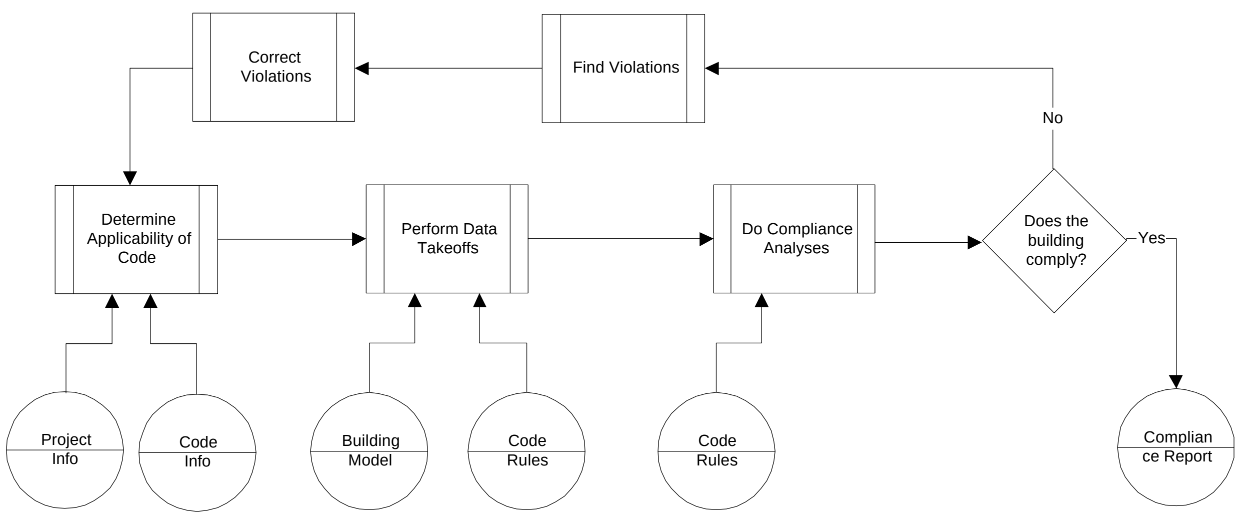
The project covers specific application of the code compliance enabling mechanism (R2_CS-1) in serving the disabled access and escape routes code compliance.

Disable access code compliance is a process of assessing whether **the access provisions and facilities** of a building complies with one or more codes or standards **that serve the needs of the wheelchair user and ambulant disabled** enforced by various codes and standards promulgation entities.

Escape route code compliance is a process of assessing whether **the exit provisions and facilities** of a building complies with one or more codes or standards **that provide safe means of escape for occupants** enforced by various codes and standards promulgation entities.

The processes are performed by building designers and code enforcement officials during early design and submission stages, respectively. Automatic code compliance software based on the IFC models created in this project will help building designers to carry out self-checking of their designs in order to detect code violations as early as possible while design changes are still relatively cheap to make. Similarly, it also help the code enforcement officials to verify the plans submitted by the designers for building approvals.

4.11.1.2 Process Diagram



4.11.1.3 Process Analysis

The process analysis is based on the outline given by the North America Chapter so as to maintain consistency. The infomation requirement relevant for most of the activities found in the process for disabled access and escape route code compliance are:

- **For Disabled Access**
- *Types of Functions of Buildings, Spaces and Accesses*
- *Space and Access Allowances*
- *Surface Requirement (materials) of Space and Access*

- *Changes in Level of Space and Access (gradient)*
- *Obstacle on Access*
- *Access Aids Provisions (symbol, handrails)*
- *Sanitary Provisions (washroom, water closet, basin, Urinals, Bath)*
- *Circulation Provisions (lift, conveyance)*
- *Transport Provisions (car park, taxi stand, boarding area)*
- *Other Provisions (Drinking Fountain, Public Telephone)*
- **For Escape Route**
- *Types and Functions of Buildings and Spaces*
- *Space Area*
- *Occupancy*
- *Type, Capacity, Location and Number of Exits*
- *Travel Distance*
- *Access to Exit*
- *Smoke Free, Fire Resistance, Sprinkler Protected Approach*
- *Area of Refuge*
- *Measurement*
- *Symbols*
- *Ventilation*

Determine Applicabitility of the Code

The building is assessed in the context of the project site, municipal, state, and federal regulations, financial requirements, insurance requirements, etc. in order to determine which codes are in force.

Input Information:
- *Project Information*
- *Code Information*

output Information:
- *Name of Code in Force*
- *Sections of Code in Force that apply*

Perform Information Takeoffs

Information is extracted from models, drawings, and specifications of the building and collected for further processing in compliance verification.

Input Information:
- *Building Model*
- *Code Takeoff Rules (how the Design & Reference Models are constructed from the Building Model)*

output Information:
- *Building Design Model (what it is)*
- *Building Reference Model (what it should be)*

Perform Compliance Analyses

Building information is analysed for whole building compliance assessments where possible. Successful completion of the analyses constitute successful compliance with the codes in force.

Input Information:
- *Building Design Model*

- *Building Reference Model*
- *Code Compliance Rules*

output Information:

- *Building Compliance Model*

Find Violations of Prescriptive Requirements

Each object and object relationship in the part of the building or system which is cast in doubt by compliance analysis is compared to each relevant requirement of the codes in force. Specific instances of violations of the codes in force are identified in the building model. Any violations of prescriptive, trade-off, or analytic requirements constitute a failure to comply with the corresponding code in force and must be corrected.

Input Information:

- *Building Compliance Model*
- *Code Rules*

output Information:

- *Code Violations Model*

Correct Violations

Using other design processes, violations are corrected. Corrections may affect the applicability of the certain sections or all of a code.

Input Information:

- *Code Violations Model*

output Information:

- *Building Model*

4.11.1.4 IFC Model Impact

New object types for disabled access

- *Ramp*
  - *- Gradient*
  - *- Material*
  - *- Width*
- *Handrail*
- *Passage*
  - *- Width*
- *CarPark*
- *ParkingLot (subclass of CarPark)*
  - *- Width*
  - *- Length*
- *SlabInCarPark*
  - *- Gradient*
  - *- Material*
- *Lift*
  - *- Width*
  - *- Lengh*
- *HandicappedSymbo*

Extensions to R1.0 object types for disabled access

- *ExternalSpace (IfcSpace)*
- *InternalSpace (IfcSpace)*
- *WaterClosetCompartment (IfcSpace)*
  - *- Width*
  - *- Lenght*
  - *- Door*
- *Washroom (IfcSpace)*
  - *- Area*
  - *- Height*
- *Floor (IfcFloor)*
  - *- Level*

New object types for escape route

- *SpaceBoundary*
  separates LIST[2:2] OF IfcSpace
- *SpaceVirtualBoundary*
- *SpacePhysicalBoundary*
- *InternalSpace*
- *ExternalSpace*
- *Access*
- *Exit*
  PartOfStorey : BuildingStorey
- *PassageWay*
  - *Area*
- *Corridor*
  - *Width*
  - *Area*
- *Obstruction*
  - *LocatedIn : Space*
- *Staircase*
  - *Width*
- *ExitStaircase*
- *Balustrade*
- *Handrail*
- *Ramp*
- *Landing*
  - *Width*
  - *Length*
- *Riser*
  - *Height*
- *Tread*
  - *Depth*
  - *Width*
- *Step*
- *SmokeStopLobby*
- *BuildingUnit*
- *Pipe*
- *Compartment*

Extensions to R1.0 object types for escape route

- *Building*
  - *Usage*
  - *Type*
  - *ContainsBasementStoreys*
  - *Sprinkler-protected*
- *Zone*
  - *Classification*
  - *Area*
  - *HasSpaceParts: LIST [1:?] OF Space*
- *Space*
  - *Function*
  - *OccupancyLoad*
  - *FloorGradient*
  - *FireHazardCategory*
  - *HasFloor: IfcFloor*
  - *IsSeparatedBy: SET [1:?] OF SpaceBoundary*
- *Opening*
  - *Area*
  - *PartOf: SpaceBoundary*

4.11.1.5 RoadMap Issues

Interoperability issues

Disciplines from which information is needed:

- Arcitectural
- HVAC

Disciplines for which information is produced:

- Arcitectural
- HVAC

Target Software Companies/Application type

- Architectural CAD software developers / Code complaince CAD system for arcitects
- Code complaince software developers / Code complaince for official regulatory bodies

Estimating and Scheduling

4.12 ES-1 - Cost Estimating

4.12.1 Object Identification

Identify selected objects in the IFC Project Model and classify them in terms of a cost estimating system. This is done by an estimator at the beginning of the process of costing the objects that will be estimated. This process uses information that was originally entered during architectural design and engineering process to make the classifications. The resulting classifications are then used by a costing system to associate objects in the drawing with objects in its database.

4.12.1.1 Industry Process Definition

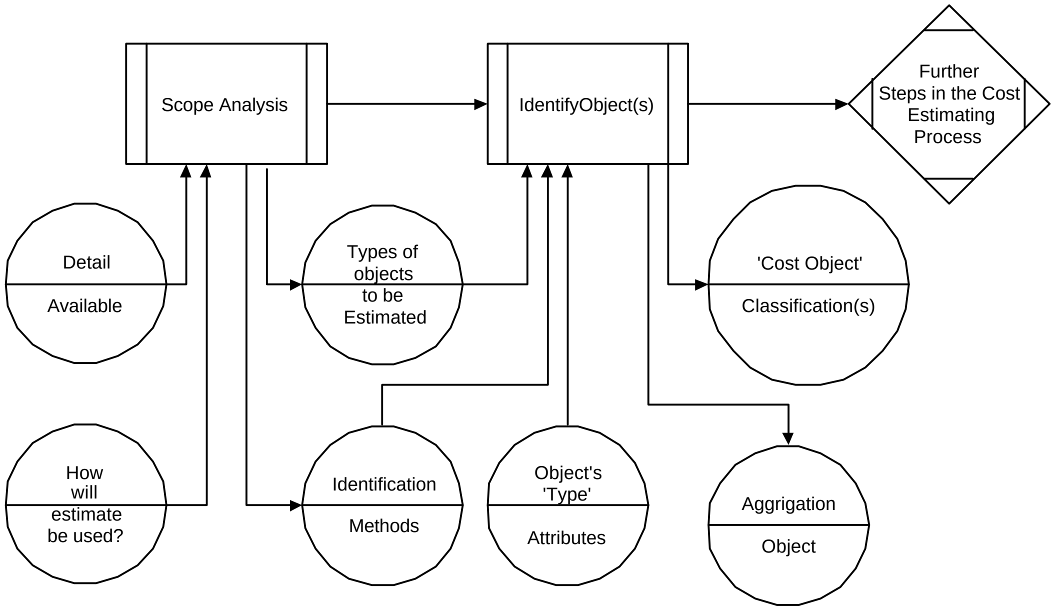
The cost estimator will first do a Scope Analysis. Its purpose is to choose what level of objects to use as the basis of the estimate and determine what information is available for creating an estimate. For instance, at the beginning of the design process, only spaces may be used. As more detail appears in the model, objects such as doors, walls, windows, etc. will be used.

After choosing the types of objects to base the estimate on, a system of identification is needed. The object's class (e.g. IfcDoor) is used to begin the identification process. Other attributes are needed to further identify the type of door. These attributes may be a classification specifications, physical attributes, dimensional information and relationships to other objects.

The estimator may need to aggregate several objects in order to make them correspond to an estimating system object. After combining these objects into an assembly, they would then classify the aggregation object in terms of the desired estimating system object. Likewise, an estimator may need to decompose an object in order to match it to estimating system objects. In this case, a set of zones that classify different parts of the object may be needed.

Once the object has been classified in terms of the cost estimating system, that classification should be recorded in the object.

4.12.1.2 Process Diagram



4.12.1.3 Process Analysis

4.12.1.3.1 Scope Analysis

The model is analyzed to determine what objects and object information is available for estimating. The purpose of the estimate is also considered.

Input Information:

- *Objects available in the model (spaces, walls, doors, manufactured parts,…)*
- *Type of information in the objects that may be used for estimating. (classifications, material specifications, dimensions…)*
- *Purpose of the estimate (conceptual, detailed, alternate, change order, basis of a bid…)*

output Information:

- *Types of objects that will be used as the basis for the estimate. For example, spaces for a conceptual estimate, or more granular objects like doors and walls for a more detailed estimate.*
- *Identification methods to be used to classify the object in terms of the cost estimating system.*

Project Model Usage Requirements:

This step is a human analysis of the state of the model. It does not required any new object types.

4.12.1.3.2 Identify Object

A class of objects is selected for estimating. All instances of that type of object are selected and classified in terms of the cost estimating system.

Input Information:

- *Types of objects to be estimated.*
- *The object's class.*
- *The object's specification according to some classification system.*
- *The object's material specification (such as wood, metal, …)*
- *The object's specification requirements (such as fire rating)*

- *The object's dimension attributes.*
- *The 'context' of the object. (for example, the material type of the wall a door installed in)*
- *The object's design status (new, changed, deleted) and version number*
- *Other attributes that may be of use for cost estimating…*

output Information:

- *Object selection criterion*
- *Aggregation of objects for estimating and/or scheduling*
- *Decomposition objects for estimating and/or scheduling*
- *The classification of the object in terms of the cost estimating system.*

Project Model Usage Requirements:

Existing Classes:

- ***Any object that may impact the cost of the project.***
  Data
  - *dimensional information*
    - *lengths, widths, volumes, …*
  - *specification information*
    - *material, functional specification, structural specification, …*
- ***IfcClassification***
  Data
  - *ClassificationPublisher -> IfcString*
    - *This references the publisher of the cost book or database.*
  - *ClassificationTable -> IfcString*
    - *This references the specific table used.*
  - *ClassificationNotation -> IfcString*
    - *This is the code for the object being classified.*
  - *ClassificationDescription -> IfcString*
    - *This is a readable description of the classification.*

New Classes:

- ***IfcAggrigation***
  Data
  - *AggrigationElements à Set [0:N] Ref IfcProductObject*
    - *This groups together objects that are estimated together.*
- ***IfcConstructionZone***
  Data
  - *ConstructionZone à IfcSpaceElement*
    - *An IfcProductObject may be decomposed into several constructions zones. For example, an IfcSlab may be decomposed into several pour zones.*

4.12.1.4 IFC Model Impact

This section summarizes the model requirements from all the process tasks analyzed above into two groups à Extensions to R1.0 model object types and proposed new object types for R2.0.

Usage/Extensions to R1.0 object types

- ***IfcProductObject***
  Data
  - *ConstructionZones à Set [0:n] ref IfcConstructionZone*
    - *The decomposition of a product object into construction zones. For example, an IfcSlab may be decomposed into several pour zones.*

New object types required

- ***IfcAggrigation***
  Data
  - *AggrigationElements à Set [0:N] Ref IfcProductObject*
    - *This groups together objects that are estimated together*
- ***IfcConstructionZone***
  Data
  - *ConstructionZone à IfcSpaceElement*
    - *An IfcProductObject may be decomposed into several constructions zones. For example, an IfcSlab may be decomposed into several pour zones.*

4.12.1.5 RoadMap Issues

Interoperability Issues

Applications from which information is needed:

- *Architecture*
- *Other disciplines that provide attribute sets to describe objects.*

Applications for which information is produced:

- *Architecture*
- *Estimating*
- *Scheduling*
- *Facilities Management*
- *Other disciplines that provide attribute sets to identify objects*

Value of software supporting this process

- *Scheduling - 1*
- *Estimating - 1*
- *Facilities Management - 3*

Sponsor Software Companies

Software Companies that have shown an interest in developing applications which implement the process

- *Timberline Software*

4.12.1.6 Issues idendified in reviews

{{ Reviewing group - Reviewed for: }}

- ***Issues:***
  - ***{{ Proposed resolution }}***
- **{{ Issue 1 }}**
  - *{{ Proposed resolution }}*
- **{{ Issue 2 }}**
  - *{{ Proposed resolution }}*

4.12.2 Task and Resource Modeling

This process models tasks and resources required to install various objects found in the project model. After an object is selected for estimating, its required installation tasks and their required resources are determined. By modeling this information, accurate estimates may be made based on material and labor costs and on predicted production rates. The tasks and resources introduced into the model at this point can be used later by the scheduling process.

4.12.2.1 Industry Process Definition

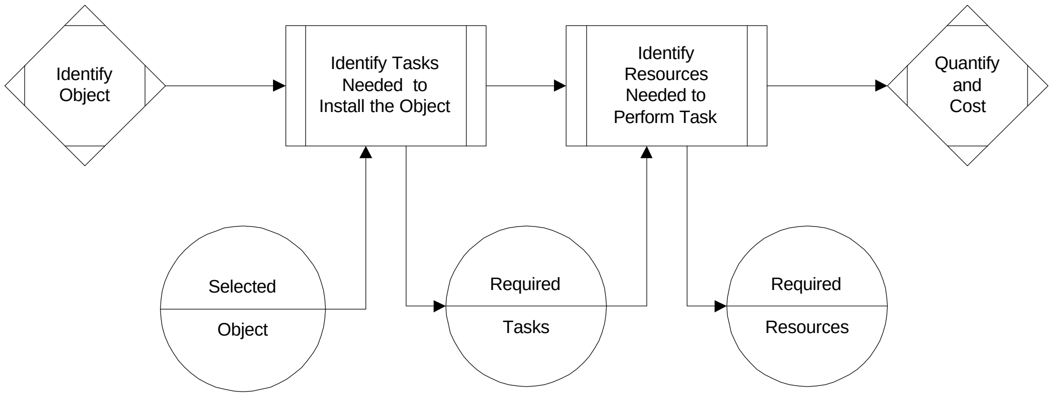
To construct or install an actual object, various tasks must be performed. For instance, to construct a wall you must ‘construct the framing’, ‘apply the sheetrock’, ‘finish the sheetrock’, and ‘apply a finish’. Each of these tasks can be modeled to estimate cost or produce a construction schedule.

Each task requires various resources. For example, framing a wall may require carpenters, lumber, and nails. The cost and availability of these resources will go into determining the cost and schedule for constructing the wall.

Selection of an appropriate construction method provides information for crew configuration and production rate, based on historical averages. These are used to calculate the duration of each task and the quantities of resources and materials.

Quantification of resources is accomplished by identifying the appropriate construction method, crew configuration, and historical production rate. These are used to calculate task durations and resource quantities.

4.12.2.2 Process Diagram



4.12.2.3 Process Analysis

4.12.2.3.1 Identify Tasks needed to install the object

The estimator examines the object to determine its construction method. The construction method will specify the tasks that need to be completed to construct the object.

Input Information:

- *Class of the object (wall, door, …)*
- *Attributes of the object (material, finish, …)*
- *Dimensions of the object (height, area…)*

output Information:

- *IfcWorkGroup - Work objects that group associated tasks.*
- *IfcWorkTask - Tasks required to construct or install the object*
- *IfcResourceObject - Resources required by work tasks..*

Project Model Usage Requirements:

Existing Classes:

- ***IfcWorkGroup***
  Data
  - *WorkGroupTitle à IfcString*
    - *This allows several tasks to be grouped together.*
  - *HasParts à Set [0:N] ref IfcWorkGroup*
    - *This allows hierarchical groupings.*
  - *ConsistsOf à Set [0:N] ref IfcWorkTask*
    - *This allows several tasks to be grouped together.*

- ***IfcWorkTask***
  Data
  - *TaskDescription à IfcAttString*
    - *Describes the task*
  - *WorkMeghod à IfcAttString*
    - *Describes the work method for the task*
  - *TotalCost à IfcCost*

- *Total cost of the task*
- *Resources à List [0:N] fcResourceObject*
  - *List of resources needed to complete the task*
- *ResourceQuantity à List [0:N] IfcAttReal*
  - *The Quantities of the above resources*
- *ResourceDuration à List [0:N] IfcAttDate*
  - *Time durations for the above resources are needed*

- ***IfcResourceObject***

  Data
  - *ResourceType à enum Labor, Equipment, Material*
    - *Specifies the basic type of resource*
  - *ResourceDescription à IfcAttString*
    - *Description of the resource. (e.g. Carpenter, Hoist, Forms.)*
  - *HasCost -> IfcUnitCost*
    - *Cost per unit*

New Classes:

- ***No new classes are required for this functionality.***

4.12.2.3.2 Identify Resources Needed to Install the Object

The estimator determines the resources required to perform each of the tasks.

Input Information:

- *IfcTask*
- *Attributes of the object (material, finish, …)*
- *Dimensions of the object (height, area…)*

output Information:

- *IfcResource*

Project Model Usage Requirements:

Existing Classes:

- ***IfcResourceObject***

  Data
  - *ResourceType à enum Labor, Equipment, Material*
    - *Specifies the basic type of resource*
  - *ResourceDescription à IfcAttString*
    - *Description of the resource. (e.g. Carpenter, Hoist, Forms.)*
  - *HasCost -> IfcUnitCost*
    - *Cost per unit*

New Classes:

- ***No new classes are required for this functionality.***

4.12.2.4 IFC Model Impact

This section summarizes the model requirements from all the process tasks analyzed above into two groups à Extensions to R1.0 model object types and proposed new object types for R2.0.

Usage/Extensions to R1.0 object types

- ***IfcWorkTask***

  Data

  - *We need to make sure that the current IfcWorkTask is able to specify any number of resource usages.*
  - *Resources à List [0:N] fcResourceObject*
    - *List of resources needed to complete the task*
  - *ResourceQuantity à List [0:N] IfcAttReal*
    - *The Quantities of the above resources*
  - *ResourceDuration à List [0:N] IfcAttDate*
    - *Time durations for the above resources are needed*

New object types required

- ***No new object types required.***

4.12.2.5 RoadMap Issues

Interoperability Issues

Applications from which information is needed:

- *Scheduling*

Applications for which information is produced:

- *Scheduling*
- *Cost Estimating*

Value of software supporting this process

{{In this section, please allow for the other domains to rank your process in order of precedance for their domain, this allows us to examine the issue on a group as well as an individual level}}

- *Estimating - 1*
- *Scheduling - 1*
- *Facilities Management - 3*

Sponsor Software Companies

Software Companies that have shown an interest in developing applications which implement the process

- *Timberline Software*

4.12.2.6 Issues idendified in reviews

{{ Reviewing group - Reviewed for: }}

- ***Issues:***
  - ***{{ Proposed resolution }}***

- **{{ Issue 1 }}**
  - *{{ Proposed resolution }}*
- **{{ Issue 2 }}**
  - *{{ Proposed resolution }}*

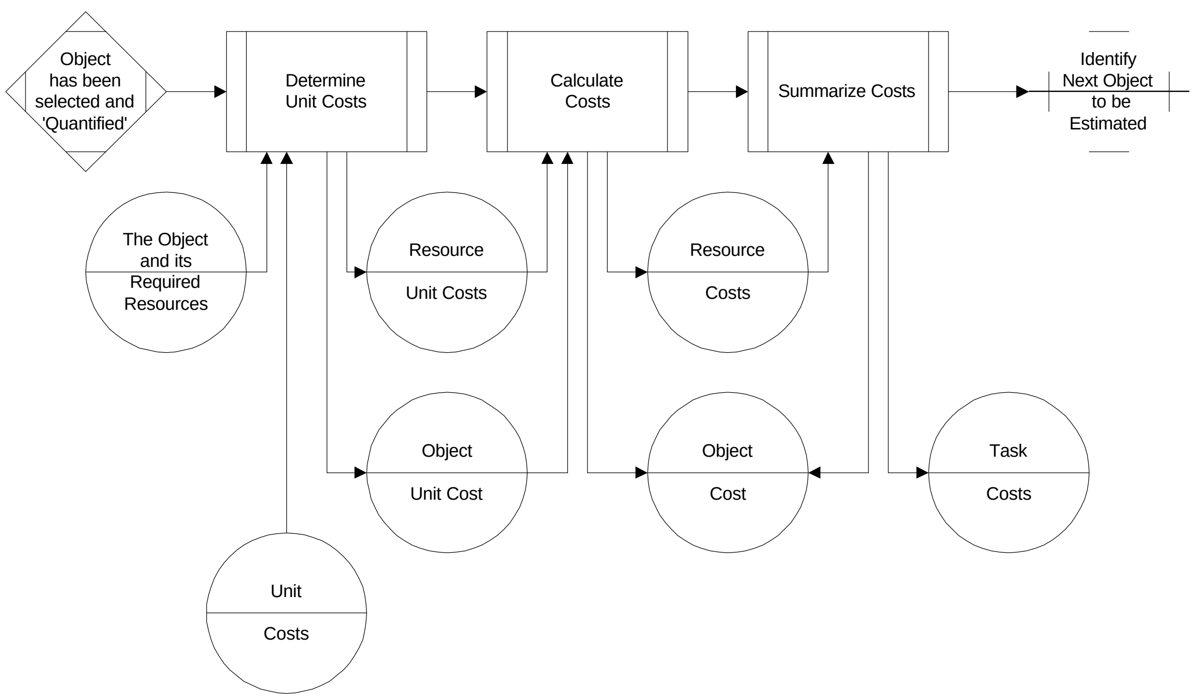
4.12.3 Cost Modeling

Once an object, its parts, and its required tasks and resources have been determined, costs may be calculated and applied to the model. This includes the unit costs of resources, the resulting costs of the tasks, the actual cost of manufactured parts, and the summarized cost of the object.

4.12.3.1 Industry Process Definition

Once an object has been selected and quantified, and its required tasks and resources have been added to the model and quantified, the object can be costed. Unit costs are determined for the various resources such as labor and materials. Alternately, in a conceptual estimate, a unit cost may be determined for the object's overall quantity (such as cost per square foot of office space). In this case, task and resource modeling has not been done, and the overall quantity must be calculated at this point. The quantities and unit costs are used to calculate the costs of the resources, tasks, and the object.

4.12.3.2 Process Diagram



4.12.3.3 Process Analysis

4.12.3.3.1 Determine Unit Costs

Once quantities have been determined for the 'overall' object or its tasks and resources, unit costs are applied. If cost is being modeled using tasks and resources, unit costs are selected for the resources. If cost is being modeled based on a unit cost for the overall object, a unit cost is select for the overall object.

Input Information:

- *Object to be costed*
- *Resources to be costed*
- *Unit costs (possibly from an estimating system or price book)*

output Information:

- *IfcUnitCost*

Project Model Usage Requirements:

Existing Classes:

- ***IfcUnitCost***

New Classes:

- ***No new classes are needed.***

4.12.3.3.2 Calculate Costs

The object and resource quantities and the selected unit costs are used to calculate the cost of the object or its resources.

Input Information:

- *Object's 'overall' quantity*
- *Resource quantities*
- *Unit costs*

output Information:

- *Resource cost (if tasks and resources are used to model the cost)*
- *Object cost (if the cost is based on the object's 'overall' quantity)*

Project Model Usage Requirements:

Existing Classes:

- ***IfcCost***
- ***IfcUnitCost***

New Classes:

- ***No new classes are needed.***

4.12.3.3.3 Summarize Costs

If an object's cost is based on the costs of its tasks and resources or on the costs of its component parts, summarize these costs at the task and object level.

Input Information:

- *Resource costs*
- *Costs of component parts*

output Information:

- *Task costs*
- *Object cost*
- *Cost schedule*

Project Model Usage Requirements:

Existing Classes:

- ***IfcWorkTask***
  - ***Data***
  - *- TaskCost à IfcCost*
    - *- Total cost of the task*
- ***IfcProductObject***
  - ***Data***
  - *- ProductCost à IfcCost*
    - *- Cost impact of the product object.*
- ***IfcCostScheduleElement***
  - ***Data***
  - *- TotalCost à IfcCost*
    - *- Total cost of this entry in the cost schedule*
  - *- UnitCost à IfcUnitCost*
    - *- Cost per unit for this entry in the cost schedule*
  - *- Description à IfcAttString*
    - *- Description for this entry in the cost schedule*
  - *- Quantity à IfcAttReal*
    - *- Quantity for this entry in the cost schedule (in terms of UnitCost unit)*

New Classes:

- ***No new classes are needed.***

4.12.3.4 IFC Model Impact

This section summarizes the model requirements from all the process tasks analyzed above into two groups à Extensions to R1.0 model object types and proposed new object types for R2.0.

Usage/Extensions to R1.0 object types

- ***IfcProductObject***
  - ***Data***
  - *- ProductCost à List[0:?] IfcCost*
    - *- We should allow for more than one cost on a product object, since it may have different costs, depending on your viewpoint. For example, is a cost an actual cost or an estimated cost?*
  - *- CostSchedule à IfcCostScheduleObject*
    - *- The IfcProductObject should be able to reference cost schedules or cost schedule elements that provide information about its cost. This allows a product object's cost to be reported in a context of other costs.*
- ***IfcCost***
  - ***Data***
  - *- CostType à Enumeration???*

- *In the current model (1.5), ProductCost is used to represent the cost of an IfcProductObject. But as mentioned above, the meaning of a cost is not precisely defined. For instance, is a ProductCost the sum of the cost of its sub-object and work objects? Is it a manufacturer's cost, or does it include cost of installation?*
- *We should specify Cost Types that would help define the meaning of an IfcCost. Some possible cost types are:*
  - *Singular cost - Costs calculated based on a unit cost and a quantity.*
  - *Aggregate cost - Sum of the costs of an object's parts.*
  - *Target cost (allowable cost?) - Used to communicate the allowable expenditure for an object. This may also be a range.*

New object types required

- ***{{ Object type name }}***

  Data
  - *{{ Data description à type }}*
    - *{{ notes }}*

  Behavior
  - *{{ Behavior description }}*
    - *{{ notes }}*

- ***No new classes are needed.***

4.12.3.5 RoadMap Issues

Interoperability Issues

Applications from which information is needed:

- *Other construction disciplines such as project management and job cost accounting.*

Applications for which information is produced:

- *Other construction disciplines which are interested in estimated costs. This includes project management and job cost accounting are interested in estimated costs.*

Value of software supporting this process

- *Estimating - 1*
- *Construction Management - 1*
- *Facilities Management - 3*

Sponsor Software Companies

Software Companies that have shown an interest in developing applications which implement the process

- *Timberline Software*

4.12.3.6 Issues idendified in reviews

{{ Reviewing group - Reviewed for: }}

- ***Issues:***
  - ***{{ Proposed resolution }}***
- **{{ Issue 1 }}**
  - *{{ Proposed resolution }}*
- **{{ Issue 2 }}**
  - *{{ Proposed resolution }}*

Facilities Management

4.13 FM-1 Engineering Maintenance

{{ This project has been delayed for inclusion in IFC Release 3.0 }}

4.14 FM-2 Architectural Mainenance

{{ This project has been delayed for inclusion in IFC Release 3.0 }}

4.15 FM-3 Property Management (Building owner's viewpoint)

4.15.1 Grouping IFC objects

This process can be performed at any stage in the lifecycle of the building, but it has been designed for the Property manager. Groups can consist of IFC Object, and the user can use the Group object to make links to the users own private objects.
The need for grouping can be caused by any management purpose, like new department, workgroup, cleaning area, renovation, fire zone etc. In this process the property manager can create new groups from selected objects. These groups can be used for any administrative or management purposes. All material or quantitative information is calculated from the IFC model. The model information can be used together with owner's own or other external database information to evaluate operational costs or other needed values.

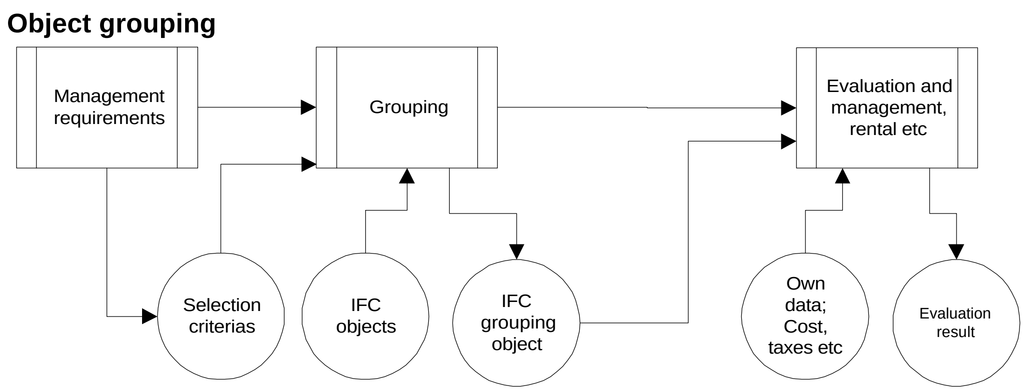
4.15.1.1 Industry Process Definition

The first task is to define the grouping purpose, which defines the classification of the group. Then the objects for new group can be selected through various methods:

- any objects selected by the user
- filtered objects (type, properties or other selection key) selected by the user
- filtered objects in the whole model

After the selection is completed the user can give a description to the group.

4.15.1.2 Process Diagram



4.15.1.3 Process Analysis

There are three processes in the Grouping IFC Objects. Manage requirements is properly supported by a Facility Management Software tool, while Grouping IFC is a IFC based application. Evaluation and other functions can be based on the users own software, it must just be able to read the needed IFC Objects.

4.15.1.3.1 Management requirements

The user makes decisions about what he would like group.

Input Information:

- *Any needs for grouping*

output Information:

- *Selection criteria*
  - *Space type either/or*
  - *Floor type*
  - *Department use of spaces*
  - *Cleaning*
  - *etc.*

Project Model Usage Requirements:

Existing Classes:

New Classes:

4.15.1.3.2 Grouping

The Grouping process reads information about what should be grouped. A number of object is selected from the IFC model either by IFC Objects type/attribute or by the user picks a number of objects from the model. A new grouping object is made and the identification of the selected objects a store/linked to the grouping object.

Input Information:

- *Section criteria*

- *IFC Objects (IFC Space, IFC Layered Element etc.)*

output Information:

- *IFC Grouping object*
  - *Identification*
  - *Description*
    - *Classification of groups*

Project Model Usage Requirements:

Existing Classes:

- ***IFC Object***

  Data

  - Identification

  - Attribute already in IFC objects

  Behavior

New Classes:

- ***IFC Grouping Object***

  Data

  - Identification

  - Description

  - Classification of group from a project specific list

  Behavior

  - {{ Behavior description }}

  - {{ notes }}

4.15.1.3.3 Management, rental etc.

Use IFC Grouping object to make connection to private database systems or to makes reports and drawings.

Input Information:

- *IFC Grouping Object*
- *Own data: Cost, taxes etc.*

Output Information:

- *Evaluation result (Rental, drawings)*

Project Model Usage Requirements:

Existing Classes:

- ***IFC Objects***

  Data

  - Identification

  - Attribute already in IFC objects

  Behavior

  - {{ Behavior description }}

  - {{ notes }}

New Classes:

- ***{{ Object type name }}***
  Data
  - *{{ Data description à type }}*
    - *{{ notes }}*

  Behavior
  - *{{ Behavior description }}*
    - *{{ notes }}*

4.15.1.4 IFC Model Impact

This section summarizes the model requirements from all the process tasks analyzed above into two groups à Extensions to R1.0 model object types and proposed new object types for R2.0.

Usage/Extensions to R1.0 object types

- ***IFC Space***
  Data
  - Room number (new)
  - Name of space (new)
  - Equipment (Electrical and HVAC) list in a string (new)

- ***IFC DoorType***
  Data
  - Thermal Rating (new)
  - Security Rating (new)
  - Change door hardware type to hardware type

- ***IFC Windowtype***
  Data
  - Thermal Rating (new)
  - Numbers of glasses (new)
  - Fire Rating for the window (new)
  - Acoustic Rating (new)
  - Security Rating (new)
  - Pointer to hardware type (new)
  - External/Internal (new ?)

- ***IFC LayeredElement***
  Data
  - Thermal Rating (new)
  - Fire Rating for the window (new)
  - Acoustic Rating (new)
  - External/Internal (new ?)

- ***IFC WallType***
  Data
  - Measure areal of an external wall ?
  - Remove Thermal Rating

- Remove Fire Rating
- Remove Acoustic Rating

New object types required

- ***IFC Grouping Object***

  Data
  - Identification
  - Description
  - Classification of group from a project specific list
  Behavior
  - {{ Behavior description }}
  - {{ notes }}

4.15.1.5 RoadMap Issues

Interoperability Issues

Applications from which information is needed:

- *Architectural (Spaces, Wall, Floors, Doors, Windows)*
- *Others in the future*

Applications for which information is produced:

- *Facility Management*
- *Any program that needs groups of objects*

Value of software supporting this process

{{In this section, please allow for the other domains to rank your process in order of precedance for their domain, this allows us to examine the issue on a group as well as an individual level}}

- *{{ discipline 1 }} - {{value from 1-10, 1 being the highest value, 10 being the lowest]]*
- *{{ discipline 2 }} - {{value from 1-10]]*

Sponsor Software Companies

Software Companies that have shown an interest in developing applications which implement the process

- *{{ company 1 }}*
- *{{ company 2 }}*

4.15.1.6 Issues identified in reviews

{{ Reviewing group - Reviewed for: }}

- ***Issues:***
  - ***{{ Proposed resolution }}***

- **{{ Issue 1 }}**
  - *{{ Proposed resolution }}*
- **{{ Issue 2 }}**
  - *{{ Proposed resolution }}*

4.15.2 Linking the maintenance objects to the IFC objects

The different building elements in the building are linked to a maintenance object. The guarantees, maintenance periods and maintenance history of these elements is stored in the maintenance object. The property manager can check from this information when maintenance operations should be done and if all necessary operations are made according to the guarantee terms. The grouping mechanism is identical to the groping activities

4.15.2.1 Industry Process Definition

First task is to define the selection criteria for a maintenance group. Then the objects for new group can be selected through various methods:

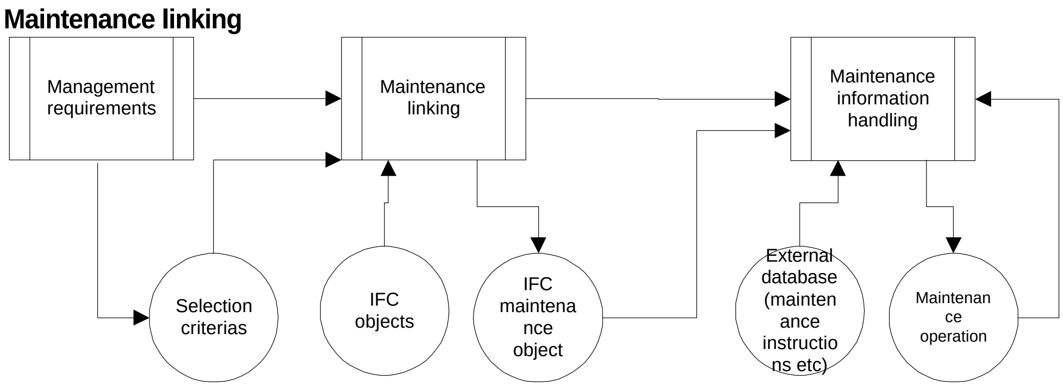
- any objects selected by the user
- filtered objects (type, properties or other selection key) selected by the user
- filtered objects in the whole model

After the selection is completed the user can give a description to the new maintenance group.
If the selected objects already belong to some maintenance group, the application should warn the user about it and ask for instructions for further operations.
When the maintenance groups are formed the user can use those as the selection criteria for different maintenance operations and reports. All maintenance data is stored in the maintenance object and the IFC object data should be available from the actual objects. The first task is to define the grouping purpose, which defines the classification of this group. Then the objects for new group can be selected through various methods:

- any objects selected by the user
- filtered objects (type, properties or other selection key) selected by the user
- filtered objects in the whole model

After the selection is completed the user can give a description to the group.

4.15.2.2 Process Diagram



4.15.2.3 Process Analysis

There are three processes in the Linking the maintenance objects to the IFC objects: Management requirements, Maintenance linking and Maintenance information handling.

4.15.2.3.1 Management requirements

The user makes decisions about what he would like maintain.

Input Information:

- *Any needs for maintenance*

output Information:

- *Selection criteria*

Project Model Usage Requirements:

Existing Classes:

New Classes:

4.15.2.3.2 Maintenance linking

IFC objects are linked to a Maintenance object. The IFC Objects are selected according to the selection criteria. A new Maintenance object is defined.

Input Information:

- *Section criteria*
- *IFC Objects (IFC Space, IFC Layered Element etc.)*

output Information:

- *IFC Maintenance object*

Project Model Usage Requirements:

Existing Classes:

- ***IFC Object***
  Data

- *ID*

Behavior

New Classes:

- ***IFC Maintenance Object***

Data

- *Identification*
- *Description*
- *Classification*
- *Maintenance period*
- *Last maintenance date*
- *Maintenance Instruction (Pointer to)*
- *Maintenance history (Pointer to)*

Behavior

- *{{ Behavior description }}*
  - *{{ notes }}*

4.15.2.3.3 Maintenance information handling

Use Maintenance Object to produce Maintenance operation schedules and instructions. Maintenance instruction are stored outside the IFC model.

Input Information:

- *IFC Maintenance Object*
- *External databases/links to maintenance instructions.*

Output Information:

- *Maintenance operations*

Project Model Usage Requirements:

Existing Classes:

- ***{{ Object type name }}***

Data

- *{{ Data description à type }}*
  - *{{ notes }}*

Behavior

- *{{ Behavior description }}*
  - *{{ notes }}*

New Classes:

- ***{{ Object type name }}***

Data

- *{{ Data description à type }}*
  - *{{ notes }}*

Behavior

- *{{ Behavior description }}*
  - *{{ notes }}*

4.15.2.4 IFC Model Impact

This section summarizes the model requirements from all the process tasks analyzed above into two groups à Extensions to R1.0 model object types and proposed new object types for R2.0.

Usage/Extensions to R1.0 object types

- ***IFC Objects (IFCWindow)***
  Data
  - Id
  - Type (IFCWindowtype)
  Behavior
  - {{ Behavior description }}
  - {{ notes }}

New object types required

- ***IFC Maintenance Object***
  Data
  - Identification
  - Description
  - Classification
  - Maintenance period
  - Last maintenance date
  - Maintenance Instruction (Pointer to)
  - Maintenance history (Pointer to)
  Behavior
  - {{ Behavior description }}
  - {{ notes }}

4.15.2.5 RoadMap Issues

Interoperability Issues

Applications from which information is needed:

- *Architectural (Wall, Doors, Windows, Floors)*
- *HVAC*

Applications for which information is produced:

- *Facility Management (Maintenance)*

Value of software supporting this process

{{In this section, please allow for the other domains to rank your process in order of precedance for their domain, this allows us to examine the issue on a group as well as an individual level}}

- *{{ discipline 1 }} - {{value from 1-10, 1 being the highest value, 10 being the lowest]]*
- *{{ discipline 2 }} - {{value from 1-10]]*

Sponsor Software Companies

Software Companies that have shown an interest in developing applications which implement the process

- *{{ company 1 }}*
- *{{ company 2 }}*

4.15.2.6 Issues identified in reviews

{{ Reviewing group - Reviewed for: }}

- ***Issues:***
    - ***{{ Proposed resolution }}***
- **{{ Issue 1 }}**
    - *{{ Proposed resolution }}*
- **{{ Issue 2 }}**
    - *{{ Proposed resolution }}*

4.16 FM-4 - Occupancy Planning (incl. Design and Layout of Workstations)

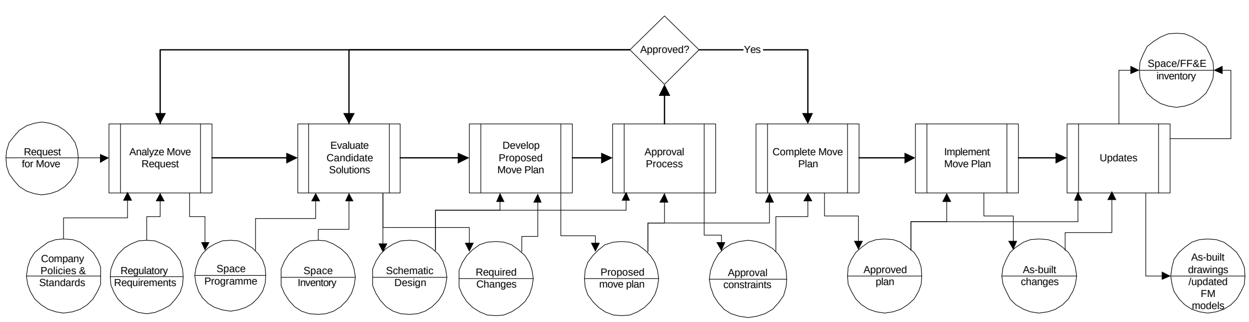
4.16.1 Occupancy Planning

The occupancy planner (includes interior designers, facilities managers, architects, furniture dealers, etc.) applies standards during the assignment of people and organizations to interior spaces. This process occurs during the initial planning of space occupancy, and whenever that occupancy needs to change (company reorganization, company growth, etc.)

4.16.1.1 Industry Process Definition

<< refer to usage scenario >>

4.16.1.2 Process Diagram



4.16.1.3 Process Analysis

4.16.1.3.1 Analyze Move Request

Evaluate request with respect to occupant information, company policies, regulatory requirements. Identify FF&E required for the occupant, and generate space programme.

Input Information:

- *Request for Move*
    - *Space area requirements*
    - *Space service requirements (cooling requirements, gases required, etc.)*
    - *Adjacencies/affinities relationships (location)*
    - *FF&E required*
    - *Department requirements*
    - *Occupant list*
    - *Target Occupancy Date*
    - *Budget*
    - *Special requirements (e.g. raised floor)*
- *Company policies and standards*
    - *Relationship between occupant position/title/department and space standard*
    - *Relationship between occupant position/title/department and equipment standard*
    - *Relationship between occupant position/title/department and furnishings standard*
- *Regulatory requirements*
    - *National or local facilities regulations (e.g. ADA, OSHA)*
    - *Local Fire/Electrical Codes*

Output Information:

- *Space programme*
    - *Number and sizes of spaces*
    - *Requested Space locations*
    - *Space characteristics*
    - *Occupant list*
    - *FF&E lists (existing/new)*
    - *Target Occupancy Date*
    - *Budget*
    - *Special requirements*

Project Model Usage Requirements:

Existing Classes:

- *IfcSpaceProgramme (referenced by IfcProgrammeGroup)*
    - {{ all attributes described in version 1.0 spec + the following }}
    - RequestedLocation: Ref. to IfcProductObject
    - <this can reference a building, storey, or space>
    - ServiceRequirements: Set [0:N] of IfcString
    - RequiredFF&E: Set [0:N] of Ref. to IfcTypeDefinition
    - TargetDate: IfcDate
    - Budget: IfcCost
    - SpecialRequirements: Set [0:N] of IfcString
- *IfcActor (referenced in IfcOccupancySchedule, IfcMoveAction)*
- *IfcControlObject (supertype of IfcPlan, IfcOccupancySchedule, IfcMoveActionConstraint)*
- *IfcCost (used in IfcPlan*
- *IfcData (used in Att_ScheduleData, IfcMoveActionConstraint)*

- *IfcElement (referenced in IfcMoveAction)*
- *IfcPlan (supertype of IfcMovePlan)*
- *IfcProcessObject (used in IfcOccupancySchedule, supertype of IfcMoveAction)*
- *IfcProgrammeGroup (referenced in IfcMovePlan)*
- *IfcProject (referenced in IfcPlan)*
- *IfcRelSequence (referenced in IfcOccupancySchedule)*
- *IfcSpace (referenced in IfcMoveAction)*
- *IfcString (used in IfcMoveActionConstraint)*
- *IfcTimeDuration (used in Att_ScheduleData)*

New classes:

- *IfcPlan ß subtype of IfcControlObject*
  ß supertype of IfcMovePlan

  {{need to varify this with Construction domain and other groups}}
  - Project: Ref. to IfcProject
  - Schedules: Set [0:N] of Ref. to IfcSchedule
  - Budget: IfcCost
  - <more to be defined>
- *Att_ScheduleData ß Extension Attributeset for any object that uses schedule data set*

  {{need to varify this with Construction domain}}
  - Total_duration: IfcTimeDuration
  - Scheduled _start_date: IfcDate
  - Scheduled_finish_date: IfcDate
  - Actual_start_date: IfcDate
  - Actual_finish_date: IfcDate
  - Early_start_date: IfcDate
  - Early_finish_date: IfcDate
  - Late_start_date: IfcDate
  - Late_finish_date: IfcDate
  - Total_float: IfcTimeDuration
  - Days_remaining: IfcTimeDuration
- *IfcOccupancySchedule ß subtype IfcControlObject*
  - Occupying_actions: Set [0:N] of Ref. to IfcProcessObject
  - <this contains a set of IfcWorktask and IfcMoveAction>
  - Pred_succs: Set [0:N] of Ref. to IfcRelSequence
  - Schedule: Att_ScheduleData
  - Responsible: Ref. to IfcActor
- *IfcMoveAction ß subtype of IfcProcessObject*
  - Occupants_to_move: Set [0:N] of Ref. to IfcActor
  - FF&E_to_move: Set [0:N] of Ref. to IfcElement
  - <these can be IfcFurniture, IfcEquipment, IfcSystemFurniture, etc.>
  - Move_from: Ref. to IfcSpace
  - Move_to: Ref. to IfcSpace
  - Schedule: Att_ScheduleData
  - Constraints: Set [0:N] of IfcMoveActionConstraint
  - Responsible: Ref. to IfcActor
- *IfcMoveActionConstraint ß subtype of IfcControlObject*

{{need to varify with Code and other groups, could possibly change to be subtype of IfcConstraint}}

- Constraint_type: IfcString
- <e.g. must be out by, etc.>
- Constraint_date: IfcDate

• *IfcMovePlan ß subtype of IfcPlan*

- OccupancySchedule: Ref. to IfcOccupancySchedule
- Worktask_schedule: Ref. to IfcWorktaskSchedule
- ProgramGroupToBeMoved: Ref. to IfcProgrammeGroup
- <this programme group references a set of Space Programs>

4.16.1.3.2 Evaluate Candidate Solutions

Compare space programme to available spaces to find candidate solutions including the changes of spaces and FF&E.

Input Information:

- *Space programme*
- *Space inventory*
  - *List of candidate spaces and characteristics (see Space Programme)*

Output Information (assuming candidate space exists):

- *Schematic Design*
  - *space assignment*
  - *schematic drawings*
- *Required changes*
  - *Space changes*
  - *FF&E changes*

Project Model Usage Requirements:

Existing Classes:

- *IfcString (used in IfcMovePlan, IfcSpaceInventory)*
- *IfcSpace (referenced by IfcSpaceInventory)*
- *IfcInteger (used in IfcSpaceInventory)*
- *IfcArea (used in IfcSpaceInventory)*

New Classes:

• *IfcMovePlan à extended from last step*

- {{ all items described in previous process steps + the following }}
- RequiredChanges: Set [0:N] of IfcString
- <each of these will become the TaskDescription for an IfcWorkTask if the move project is approved and implemented>
- Documents: Set [0:N] of Ref. to IfcDocument (to be defined in the next process)
- <e.g. schemetic drawings, etc.>

• *IfcSpaceInventory*

- Inventory_description: IfcString
- Spaces: Set [0:N] of Ref. to IfcSpace
- Total_Spaces: IfcInteger
- Total_NetArea: IfcArea

4.16.1.3.3 Develop Proposed Move Plan

During the design and generation of drawings, we allow for client review and approval. Define temporary staging areas, generate schedules, identify sources of all FF&E required and generate a cost estimate.

Input Information:

- *Schematic design*
- *Required changes*

Output Information:

- *Proposed move plan*
  - *Drawing*
  - *Schedule*
  - *Cost estimate*

Project Model Usage Requirements:

Existing Classes:

- *IfcCostSchedule (used in IfcMovePlan)*

New Classes:

- *IfcMovePlan à extended from last step*
  - {{ all items described in previous process steps + the following }}
  - ProjectCostEstimate: IfcCostSchedule

4.16.1.3.4 Approval Process

Occupant and management review proposed move plan and either approve (possibly with constraints) or rejects --> revert to previous steps.

Input Information:

- *Proposed move plan*
  - *Drawing*
  - *Schedule*
  - *Cost estimate*

Output Information:

- *Approval constraints*
  - *Limitations on move plan*

Project Model Usage Requirements:

Existing Classes:

- *IfcString (used in IfcMovePlan)*

New Classes:

- *IfcMovePlan à extended from last step*
  - {{ all items described in previous process steps + the following }}
  - ApprovalConstraints: Set [0:N] of IfcString

  {{Approval is subject to these added constraints – interpretation is left to FM application}}

4.16.1.3.5 Complete Move Plan

Modify proposed plan to comply with constraints. Generate work orders and purchase orders.

Input Information:

- *Proposed move plan*
- *Approval constraints*

Output Information:

- *Approved plan*
  - *record drawing set*
  - *move schedule*
  - *installation schedule*
    - *work orders*
  - *purchase orders*

Project Model Usage Requirements:

Existing Classes:

- *IfcString (used in IfcPurchaseOrder, IfcPurchaseOrderItem)*
- *IfcActor (referenced in IfcPurchaseOrder)*
- *IfcDate (referenced in IfcPurchaseOrder)*
- *IfcCost (used in IfcPurchaseOrder)*
- *IfcInteger (used in IfcPurchaseOrder, IfcPurchaseOrderItem)*
- *IfcReal (used in IfcPurchaseOrderItem)*
- *IfcUnit (used in IfcPurchaseOrderItem)*

New Classes:

- *IfcMovePlan à extended from last step*
  {{ all items described in previous process steps + the following }}
  - WorkOrders: Set [0:N] of Ref. to IfcWorkOrder
    {{ List of references to work orders necessary to complete the Occupancy Schedule}}
  - PurchaseOrders: Set [0:N] of Ref. of IfcPurchaseOrder
    {{ List of references to purchase orders necessary to complete the Occupancy Schedule}}
- *IfcWorkOrder*
  <to be defined>
- *IfcPurchaseOrder*
  - Purchase_order_No: IfcString
  - Company_title: Ref. to IfcActor
  - Supplier_name: Ref. to IfcActor
  - Date: IfcDate
  - Remark: Set [0:N] of IfcString
  - Date_required: IfcString
  - Date_scheduled: IfcDate
  - Date_actual: IfcDate
  - FOB: IfcString
  - Ship_method: IfcString
  - Total_cost: IfcCost
  - Total_items: IfcInteger
  - Purchase_items: List [0:N] of IfcPurchaseOrderItem
- *IfcPurchaseOrderItem*
  - Item_number: IfcInteger

- Quantity: IfcReal
- Code: IfcString
- Unit: IfcUnit
- Unit_price: IfcCost
- Total_cost: IfcCost
- Invoice_amount: IfcCost
- Total_balance: IfcCost
- In_PurchaseOrder: Ref. to IfcPurchaseOrder

4.16.1.3.6 Implement Move Plan

Purchase FF&E. Perform work orders. Deal with change orders. Move the occupant.

Input Information:

- *Approved plan*

Output Information:

- *As-built change*

  <change notes for drawings and documents>

Project Model Usage Requirements:

Existing Classes:

- *IfcString (used by IfcChangeOrder)*
- *IfcDate (used by IfcChangeOrder)*
- *IfcActor (referenced by IfcChangeOrder)*

New Classes:

- *IfcMovePlan* à *extended from last step*
  - {{ all items described in previous process steps + the following }}
  - ChangeOrders: Set [0:N] of Ref. to IfcChangeOrder
  - <set of references to change orders to accomplish adjustments to the Occupancy Schedule>
- *IfcChangeOrder*
  - Change_order_No: IfcString
  - Description: Set [0:N] of IfcString
  - Date: IfcDate
  - Issued_by: Ref. to IfcActor
  - Issued_to: Ref. to IfcActor
  - <more to be defined>

4.16.1.3.7 Updates

Revised documentation and databases to reflect new and revised spaces and assets.

Input Information:

- *Approved plan – as modified through the implementation*
- *As-built changs*
- *Space/FF&E inventory*

Output Information:

- *As-built drawings /updated FM models*
- *updated space/FF&E inventory*

Project Model Usage Requirements:

Existing Classes:

- *IfcString (used in IfcFurnitureInventory, IfcEquipmentInventory)*
- *IfcCost (used in IfcFurnitureInventory, IfcEquipmentInventory)*
- *IfcDate (used in IfcFurnitureInventory, IfcEquipmentInventory)*
- *IfcElement (used in IfcFurnitureInventory)*
- *IfcEquipment (used in IfcEquipmentInventory)*

New Classes:

- *IfcFurnitureInventory*
  - General_description: IfcString
  - Totol_value: IfcCost
  - Last_update_date: IfcDate
  - Furniture_inventory: Set [0:N] of IfcElement
  - <contains set of IfcFurniture and IfcWorkstation>
- *IfcEquipmentInventory*
  - General_description: IfcString
  - Totol_value: IfcCost
  - Last_update_date: IfcDate
  - Equipment_inventory: Set [0:N] of IfcEquipment

4.16.1.4 IFC Model Impact

Usage/Extensions to R1.0 object types

- *IfcSpaceProgramme (referenced by IfcProgrammeGroup)*
  - {{ all attributes described in version 1.0 spec + the following }}
  - RequestedLocation: Ref. to IfcProductObject
  - <this can reference a building, storey, or space>
  - ServiceRequirements: Set [0:N] of IfcString
  - RequiredFF&E: Set [0:N] of Ref. to IfcTypeDefinition
  - TargetDate: IfcDate
  - Budget: IfcCost
  - SpecialRequirements: Set [0:N] of IfcString
- *IfcActor (referenced in IfcOccupancySchedule, IfcMoveAction, IfcChangeOrder, IfcPurchaseOrder)*
- *IfcArea (used in IfcSpaceInventory)*
- *IfcControlObject (supertype of IfcPlan, IfcOccupancySchedule, IfcMoveActionConstraint)*
- *IfcCost (used in IfcPlan, IfcPurchaseOrder, IfcFurnitureInventory, IfcEquipmentInventory)*
- *IfcCostSchedule (used in IfcMovePlan)*
- *IfcData (used in Att_ScheduleData, IfcMoveActionConstraint, IfcPurchaseOrder, IfcChangeOrder, IfcFurnitureInventory, IfcEquipmentInventory)*
- *IfcElement (referenced in IfcMoveAction, IfcFurnitureInventory)*
- *IfcEquipment (used in IfcEquipmentInventory)*
- *IfcInteger (used in IfcPurchaseOrder, IfcPurchaseOrderItem, IfcSpaceInventory)*

- *IfcPlan (supertype of IfcMovePlan)*
- *IfcProcessObject (used in IfcOccupancySchedule, supertype of IfcMoveAction)*
- *IfcProgrammeGroup (referenced in IfcMovePlan)*
- *IfcProject (referenced in IfcPlan)*
- *IfcReal (used in IfcPurchaseOrderItem)*
- *IfcRelSequence (referenced in IfcOccupancySchedule)*
- *IfcSpace (referenced by IfcSpaceInventory, IfcMoveAction)*
- *IfcString (used by IfcMovePlan, IfcChangeOrder, IfcFurnitureInventory, IfcEquipmentInventory, IfcMoveActionConstraint, IfcMovePlan, IfcSpaceInventory, IfcPurchaseOrder, IfcPurchaseOrderItem)*
- *IfcTimeDuration (used in Att_ScheduleData)*
- *IfcUnit (used in IfcPurchaseOrderItem)*

New object types required

- *IfcDocument (used in IfcMovePlan but defined in the next process)*
- *IfcPlan ß subtype of IfcControlObject*
  ß supertype of IfcMovePlan

  {{need to varify this with Construction domain and other groups}}
  - Project: Ref. to IfcProject
  - Schedules: Set [0:N] of Ref. to IfcSchedule
  - Budget: IfcCost
  - <more to be defined>
- *Att_ScheduleData ß Extension Attributeset for any object that uses schedule data set*

  {{need to varify this with Construction domain}}
  - Total_duration: IfcTimeDuration
  - Scheduled _start_date: IfcDate
  - Scheduled_finish_date: IfcDate
  - Actual_start_date: IfcDate
  - Actual_finish_date: IfcDate
  - Early_start_date: IfcDate
  - Early_finish_date: IfcDate
  - Late_start_date: IfcDate
  - Late_finish_date: IfcDate
  - Total_float: IfcTimeDuration
  - Days_remaining: IfcTimeDuration
- *IfcOccupancySchedule ß subtype IfcControlObject*
  - Occupying_actions: Set [0:N] of Ref. to IfcProcessObject
  - <this contains a set of IfcWorktask and IfcMoveAction>
  - Pred_succs: Set [0:N] of Ref. to IfcRelSequence
  - Schedule: Att_ScheduleData
  - Responsible: Ref. to IfcActor
- *IfcMoveAction ß subtype of IfcProcessObject*
  - Occupants_to_move: Set [0:N] of Ref. to IfcActor
  - FF&E_to_move: Set [0:N] of Ref. to IfcElement
  - <these can be IfcFurniture, IfcEquipment, IfcSystemFurniture, etc.>
  - Move_from: Ref. to IfcSpace

- Move_to: Ref. to IfcSpace
- Schedule: Att_ScheduleData
- Constraints: Set [0:N] of IfcMoveActionConstraint
- Responsible: Ref. to IfcActor

• *IfcMoveActionConstraint ß subtype of IfcControlObject*

{{need to varify with Code and other groups, could possibly change to be subtype of IfcConstraint}}

- Constraint_type: IfcString
- <e.g. must be out by, etc.>
- Constraint_date: IfcDate

• *IfcMovePlan ß subtype of IfcPlan*

- OccupancySchedule: Ref. to IfcOccupancySchedule
- Worktask_schedule: Ref. to IfcWorktaskSchedule
- ProgramGroupToBeMoved: Ref. to IfcProgrammeGroup
- <this programme group references a set of Space Programs>
- RequiredChanges: Set [0:N] of IfcString
- <each of these will become the TaskDescription for an IfcWorkTask if the move project is approved and implemented>
- Documents: Set [0:N] of Ref. to IfcDocument (to be defined in the next process)
- <e.g. schemetic drawings, etc.>
- ProjectCostEstimate: IfcCostSchedule
- ApprovalConstraints: Set [0:N] of IfcString

{{Approval is subject to these added constraints – interpretation is left to FM application}}

- WorkOrders: Set [0:N] of Ref. to IfcWorkOrder

{{ List of references to work orders necessary to complete the Occupancy Schedule}}

- PurchaseOrders: Set [0:N] of Ref. of IfcPurchaseOrder

{{ List of references to purchase orders necessary to complete the Occupancy Schedule}}

- ChangeOrders: Set [0:N] of Ref. to IfcChangeOrder
- <set of references to change orders to accomplish adjustments to the Occupancy Schedule>

• *IfcSpaceInventory*

- Inventory_description: IfcString
- Spaces: Set [0:N] of Ref. to IfcSpace
- Total_Spaces: IfcInteger
- Total_NetArea: IfcArea

• *IfcWorkOrder*

<to be defined>

• *IfcPurchaseOrder*

- Purchase_order_No: IfcString
- Company_title: Ref. to IfcActor
- Supplier_name: Ref. to IfcActor
- Date: IfcDate
- Remark: Set [0:N] of IfcString
- Date_required: IfcString
- Date_scheduled: IfcDate
- Date_actual: IfcDate
- FOB: IfcString
- Ship_method: IfcString
- Total_cost: IfcCost
- Total_items: IfcInteger
- Purchase_items: List [0:N] of IfcPurchaseOrderItem

- *IfcPurchaseOrderItem*
  - Item_number: IfcInteger
  - Quantity: IfcReal
  - Code: IfcString
  - Unit: IfcUnit
  - Unit_price: IfcCost
  - Total_cost: IfcCost
  - Invoice_amount: IfcCost
  - Total_balance: IfcCost
  - In_PurchaseOrder: Ref. to IfcPurchaseOrder
- *IfcChangeOrder*
  - Change_order_No: IfcString
  - Description: Set [0:N] of IfcString
  - Date: IfcDate
  - Issued_by: Ref. to IfcActor
  - Issued_to: Ref. to IfcActor
  - <more to be defined>
- *IfcFurnitureInventory*
  - General_description: IfcString
  - Totol_value: IfcCost
  - Last_update_date: IfcDate
  - Furniture_inventory: Set [0:N] of IfcElement
  - <contains set of IfcFurniture and IfcWorkstation>
- *IfcEquipmentInventory*
  - General_description: IfcString
  - Totol_value: IfcCost
  - Last_update_date: IfcDate
  - Equipment_inventory: Set [0:N] of IfcEquipment

4.16.1.5 RoadMap Issues

Interoperability Issues

Applications from which information is needed:

- *Architecture (Spaces, Walls)*
- *Electrical (Wiring, cabling)*
- *Communications (Telco, networks)*
- *HVAC System (cooling capacity, airflow, humidity, etc.)*

Applications for which information is produced:

- *Architecture (as-builts)*
- *Electrical (Wiring, cabling)*
- *Communications (Telco, networks)*
- *HVAC System (cooling capacity, airflow, humidity, etc.)*

Value of software supporting this process

- *FM: Very High (in the top 3)*
- *Architecture: High (in the top 5)*
- *CM/Cost: Very High (in the top 3)*
- *Building Service:*

- *HVAC:*

Sponsor Software Companies

Software Companies that have shown an interest in developing applications which implement the process

- *Naoki Systems Inc.*

4.16.1.6 Issues idendified in reviews

{{ Reviewing group - Reviewed for: }}

- ***Issues:***
    - ***{{ Proposed resolution }}***
- **{{ Issue 1 }}**
    - *{{ Proposed resolution }}*
- **{{ Issue 2 }}**
    - *{{ Proposed resolution }}*

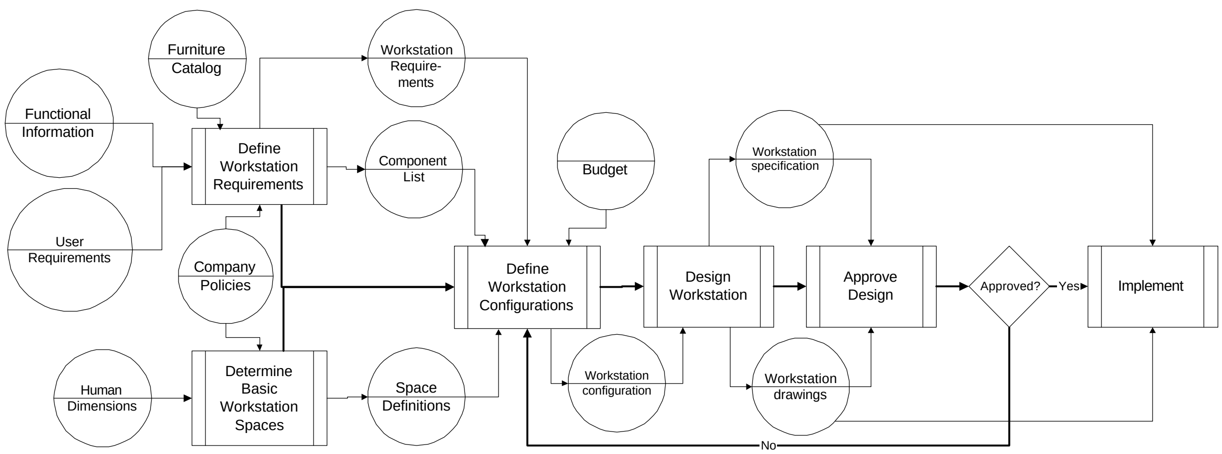
4.16.2 Design of Workstations

The facility manager (also interior designers, architects, furniture manufactures and designers, etc.) designs typical workstations to be used by office staff. The workstations designed will be selected in the layout of the systems furniture. This is a crucial process in the entire systems furniture design and planning for an open office.

4.16.2.1 Industry Process Definition

{{The process describes the steps involved in designing a single workstation of systems furniture for an open or modified office. More information refers to usage scenario.}}

4.16.2.2 Process Diagram



4.16.2.3 Process Analysis

4.16.2.3.1 Define Workstation Requirements

Define the basic components, component types and equipment, and the security, privacy and special requirements according to the employee type, work types, and company policies, etc..

Input Information:

- *Functional information*
    - *Employee Information*
        - *Employee type*
            - *secretarial, managerial, technical, programmer, reception, telephone respondent, etc.*
        - *Number of employees who share the workstation*
    - *Task Information*
        - *Work type*
            - word processing, programming, drafting, etc.
        - *Work load*
            - e.g. average of weekly working hours
            - amount of paper files to be stored
            - average amount of files produced daily
- *User requirements*

  - *special requirements for the particular user, e.g. a wheelchair, special size requirement of the chairs.*
- *Company policies*
  - *Max. and min. for managers or employees*

Output Information:

- *Component List*
  - *Workstation component types (all following items include dimensions)*
    - *Work surfaces (including size)*
      - writing & reading surface
      - computer surface
      - meeting surface
      - conference surface
      - reference material surface
      - printer surface
    - *Storage and storage types*
      - overhead storage
      - shelf storage
      - stationary storage
      - office supplies
      - personal items
      - drawers
      - file storage (file cabinets, file trays)
    - *Light fixtures*
      - ceiling-mounted lighting
      - task lighting
    - *Panels*
      - workstation partitions (glazed partitions, partitions with door, curved partitions)
      - screens (solid screen, glass screen)
    - *Seating*
      - guest seating (number of seats)
      - desk seating (number of seats)
  - *List of equipment types*
    - *Telephone, answering machine*
    - *electric typewriter*
    - *Computer, external modem, external CD-ROM, tape drive, speakers, etc.*
    - *Printer, fax machine, copier, scanner*
- *Workstation requirements*
  - *Security requirements*
    - *drawers must lock*
    - *files must lock*
    - *fireproof file cabinet*
    - *password required to access computer*
  - *Privacy requirements*
    - *speech privacy*
    - *visual privacy*
    - *equipment sharing (sharable or not)*
  - *Special requirements*
    - *face-to-face interaction*

- *files delivery requirements*
- *aesthetic requirements*

Project Model Usage Requirements:

Existing Classes:

- *IfcActor (used by IfcWorkstation)*
- *IfcArea (used in IfcWorkstationCompanyPolicy)*
- *IfcAssembledElement (superclass of IfcWorkstation)*
- *IfcBoolean (used by Att_Storage) <not sure whether IfcBoolean exists though)*
- *IfcBoundingBox (used by IfcWorkstation)*
- *IfcControlObject (superclass of IfcWorkstationWorkload, IfcWorkstationRequirement, IfcWorkstationCompanyPolicy)*
- *IfcCost (used by IfcWorkstationCompanyPolicy)*
- *IfcFixture (used by Att_TaskLighting)*
- *IfcInteger (used by IfcWorkstationWorkload, Att_Storage)*
- *IfcLength (used by Att_SystemFurnitureType, Att_Worksurface, Att_Panel)*
- *IfcManufacturedElement (supertype of IfcSystemFurniture, used in IfcWorkstation)*
- *IfcOfficeEquipment (used by IfcWorkstation)*
- *IfcPolyCurve2D (used by IfcWorkstation, Att_Panel)*
- *IfcReal (used by IfcWorkstationWorkload)*
- *IfcSpace (referenced in IfcSystemFurniture)*
- *IfcString (used by IfcWorkstationRequirement, IfcWorkstationCompanyPolicy, IfcWorkstation, Att_SystemFurnitureType, Att_Panel, Att_Worksurface, Att_Storage)*
- *IfcTimeDuration (used by IfcWorkstationWorkload)*
- *IfcTypeDefinition (used by IfcSystemFurniture, Panel, Worksurface, Storage)*
- *Att_FurnitureType*
  - <attributes definied in IFC 1.0 spec + following>
  - Product_code: IfcString
  - Width: IfcLength <nominal overall width>
  - Height: IfcLength <nominal overall height>
  - Depth: IfcLength <nominal overall depth>
  - Material: IfcString
  - Finishing: IfcString <e.g. walnut, fabric>

New classes:

- *IfcWorkstationWorkload ß subclass of IfcControlObject*
  - Average_workhour_weekly: IfcTimeDuration
  - Total_paperfiles_stored: IfcInteger
  - <used to determine file storage>
  - Average_paperfiles_produced_daily: IfcInteger
  - Total_computerfiles: IfcReal
  - <used to determine computer equipment>
  - <in unit of MB>

- *IfcWorkstationCompanyPolicy ß subclass of IfcControlObject*
  - Employee_Type: IfcString
  - <e.g. manager, programmer, secretary, etc.>
  - Max_workstation_size: IfcArea
  - Min_workstation_size: IfcArea
  - Furniture_style: IfcString
  - Cost_limit: IfcCost
- *IfcWorkstation ß subclass of IfcAssembledElement*
  - Components: Set [0:N] of Ref. to IfcManufacturedElement
  - <list of worksurfaces and storages, tables, chairs, etc., excluding the vertical panels>
  - Equipment: Set [0:N] of Ref. to IfcOfficeEquipment
  - <list of office equipment>
  - Panels: Set [0:N] of Ref. to IfcSystemFurniture
  - <list of furniture types, i.e. panel types>
  - Profile: IfcPolyCurve2D
  - Group: Ref. to IfcWorkstationGroup (to be defined in the next process)
  - <represents the workstation group that the workstation belongs to >
  - Has_boundingbox: IfcBoundingBox
  - Workload: IfcWorkstationWorkload
  - Company_policy: Ref. to IfcWorkstationCompanyPolicy
  - Assigned_to: Set [0:N] of Ref. to IfcActor
- *IfcSystemFurniture ß subtype of IfcManufacturedElement*
  - Furniture_type: Ref. to IfcTypeDefinition
  - <Panel, Worksurface, Storage>
  - Workstation: Set [0:N] of Ref. to IfcWorkstation
  - Stored_in: Ref. to IfcSpace
- *Att_SystemFurnitureType ß shared AttributeSet of Furniture_type in IfcSystemFurniture*
  - Group_code: IfcString <e.g. panels, worksurfaces, storages, etc.>
  - Width: IfcLength <i.e. nominal width>
  - Height: IfcLength <i.e. nominal length>
  - Finishing: IfcString <e.g. walnut, fabric>
- *Panel ß of IfcTypeDefinition of Type in IfcSystemFurniture*

  target object: IfcSystemFurniture

  shared = Att_SystemFurnitureType

  occurance = Att_Panel
- *Worksurface ß of IfcTypeDefinition of FurnitureType in IfcSystemFurniture*

  target object: IfcSystemFurniture

  shared = Att_SystemFurnitureType

  occurance = Att_Worksurface
- *Storage ß of IfcTypeDefinition of FurnitureType in IfcSystemFurniture*

  target object: IfcSystemFurniture

  shared = Att_SystemFurnitureType

  occurance = Att_Storage
- *Att_Panel ß occurance AttributeSet for FurnitureType Panel in IfcSystemFurniture*
  - Shape: IfcPolyCurve2D
  - Opening: IfcPolyCurve2D
  - Panel_type: IfcString
  - <e.g. Acoustical, Horz_Seg, Monolithic, Glazed, Open, Ends, Door, Screen, etc.>
  - Thickness: IfcLength
- *Att_Worksurface ß occurance AttributeSet for FurnitureType Worksurface in IfcSystemFurniture*

- Use_Purpose: IfcString
- <e.g. writing/reading, computer, meeting, printer, reference files, etc.>
- Support_type: IfcString
- <i.e. Freestanding or supported>
- Hunging_Height: IfcLength
- Thickness: IfcLength
- Shape_description: IfcString
- <corner square, rectangle, etc.>

- *Att_Storage ß occurance AttributeSet for FurnitureType Storage in IfcSystemFurniture*
  - IsOverhead: IfcBoolean (not sure whether IfcBoolean exists in the current version)
  - Support_type: IfcString
  - <i.e. Freestanding or supported>
  - Use_Purpose: IfcString
  - <e.g. shelf, stationary, office supplies, personal items, etc.>
  - Number_of_drawers: IfcInteger
  - Hunging_height: IfcLength <if IsOverhead>
  - Depth: IfcLength
- *Att_TaskLighting ß occurance AttributeSet in IfcFixture*
  - <<to be defined>>
- *IfcWorkstationRequirement ß subclass of IfcControlObject*
  - Security_Requirements: Set [0:N] of IfcString
  - Privacy_Requirements: Set [0:N] of IfcString
  - Special_Requirements: Set [0:N] of IfcString

4.16.2.3.2 Determine Basic Workstation Spaces

Define spaces of the workstation (including circulation space inside of the workstation) according to the basic requirement of human dimension standards, and company policies.

Input Information:

- *Human dimension standards (width and height)*
  - *worktask zone*
  - *sitting zone and chair clearance zone*
  - *turnaround zone*
- *Company policies*

Output Information:

- *Space definitions*
  - *circulation space*
  - *workstation space dimension*

Project Model Usage Requirements:

Existing Classes:

- *IfcString (used in IfcWorkstationZone2D)*
- *IfcLength (used in IfcWorkstationZone2D)*

New Classes:

- *IfcWorkstationZone2D ß used by IfcWorkstation*
  - <this class has no superclass>
  - Workstation_zonetype: IfcString
  - <e.g. worktask, cirrculation, chair_clearance, etc.>
  - Length: IfcLength

- Width: IfcLength
- In_workstation: Ref. to IfcWorkstation

- *IfcWorkstation*
    - <attributes defined in previous steps + the following>
    - Zones: Set [0:N] of IfcWorkstationZone2D

4.16.2.3.3 Define Workstation Configurations

Finalize all workstation components with all detailed dimensions and material information, and spaces.

Input Information:

- *Workstation information*
- *Circulation space zone*
- *Workstation requirements*

Output Information:

- *Workstation configurations*
    - *list of workstation components*
        - *types*
        - *dimensions*
        - *materials*
    - *list of equipment*
        - *types*
        - *brands*
        - *dimensions*

Project Model Usage Requirements:

Existing Classes:

- *none from this step*

New Classes:

- *none from this step*

4.16.2.3.4 Design Workstation

Produce the workstation drawings and define the specifications according to the configurations.

Input Information:

- *Workstation configurations*

Output Information:

- *Workstation layout drawings*
    - *drawing id, drawing title, author, proof, company, etc.*
- *Workstation design specifications*
    - *Materials*
    - *Installation requirements*
    - *category id, category name, item id, item name, item description, etc.*

Project Model Usage Requirements:

Existing Classes:

- *IfcControlObject (superclass of IfcDocument)*
- *IfcDate (used in Att_ElectronicDocument, Att_DocumentType)*
- *IfcInteger (used in Att_PaperDocument, Att_DocumentType, Att_Specification)*
- *IfcProductObject (superclass of IfcDocument, used by Att_PaperDocument)*
- *IfcReal (used in Att_ElectronicDocument, Att_Drawing)*
- *IfcString (used in Att_ElectronicDocument, Att_DocumentType, Att_Drawing, Att_Specification)*
- *IfcTime (used in Att_ElectronicDocument)*
- *IfcTypeDefinition (used in IfcDocument)*
- *IfcUnit (used in Att_Drawing)*

New Classes:

- *IfcDocument ß subclass of IfcControlObject*
  ß subclass of IfcProductObject
  - GenericDocumentType: IfcTypeDefinition
  - <used to differentiate between an electronic and a paper document>
  - DocumentType: IfcTypeDefinition
  - <used to differentiate between a drawing and specification, and etc.>
- *ElectronicDocument ß of IfcTypeDefinition of GenericDocumentType in IfcDocument*
  target object = "IfcDocument"
  shared = <none>
  occurance = Att_ElectronicDocument
- *PaperDocument ß of IfcTypeDefinition of GenericDocumentType in IfcDocument*
  target object= "IfcDocument"
  shared = <none>
  occurance = Att_PaperDocument
- *Att_ElectronicDocument ß occurance AttributeSet for GenericDocumentType ElectronicDocument in IfcDocument*
  - File_name: IfcString
  - FileExtension_name: IfcString
  - Software: IfcString
  - File_size: IfcReal (in unit of KB)
  - Directory: IftString
  - Backup_file: Ref. to IfcDocument
  - Paper_copy: Ref. to IfcDocument
  - Last_save_time: IfcTime
  - Last_save_date: IfcDate
  - Type: IfcString <hiden, readonly, etc.)
- *Att_PaperDocument ß occurance AttributeSet for GenericDocumentType Paper Document in IfcDocument*
  - Location: Ref. to IfcProductObject <more appropriate if there is something like IfcRoot>
  - Total_pages: IfcInteger
  - Electronic_copy: Ref. to IfcDocument
- *Drawing ß of IfcTypeDefinition of DocumentType in IfcDocument*
  target object = "IfcDocument"
  shared = Att_DocumentType
  occurance = Att_Drawing

- *Specification ß of IfcTypeDefinition of DocumentType in IfcDocument*
  target object = “IfcDocument”
  shared = Att_DocumentType
  occurance = Att_Specification
- *Att_DocumentType ß shared AttributeSet for DocumentType in IfcDocument*
  - Author: IfcString
  - Company: IfcString
  - Title: IfcString
  - Revision_Code: IfcString
  - Revision_Number: IfcInteger
  - Last_modified_date: IfcDate
  - First_created_date: IfcDate
- *Att_Drawing ß occurance AttributeSet for DocumentType Drawing in IfcDocument*
  - Drawin_id: IfcString
  - Specifications: Set [0:N] of Ref. to IfcDocument
  - Scale: IfcReal
  - Unit: IfcUnit
  - Related_drawings: Set [0:N] of Ref. to IfcDocument
- *Att_Specification ß occurance AttributeSet for DocumentType Specificatioin in IfcDocument*
  - Specification_id: IfcString
  - General_description: IfcString
  - Related_drawings: Set [0:N] of Ref. to IfcDocument
  - Total_words: IfcInteger

4.16.2.3.5 Approve Design

The process examines the design and attempts to approve it.

Input Information:

- *Workstation layout drawings*
- *Workstation design specifications*

Output Information:

- *approved design*
  - *<< the drawings and specs from input that are approved >>*

Project Model Usage Requirements:

Existing Classes:

- *none from this step*

New Classes:

- *none from this step*

4.16.2.4 IFC Model Impact

Usage/Extensions to R1.0 object types

- *Att_FurnitureType*
  - <attributes definied in IFC 1.0 spec + following>
  - Product_code: IfcString
  - Width: IfcLength <nominal overall width>
  - Height: IfcLength <nominal overall height>
  - Depth: IfcLength <nominal overall depth>

- Material: IfcString
- Finishing: IfcString <e.g. walnut, fabric>

- *IfcActor (used by IfcWorkstation)*
- *IfcArea (used in IfcWorkstationCompanyPolicy)*
- *IfcAssembledElement (superclass of IfcWorkstation)*
- *IfcBoolean (used by Att_Storage) <not sure whether IfcBoolean exists though)*
- *IfcBoundingBox (used by IfcWorkstation)*
- *IfcControlObject (IfcDocument, IfcWorkstationWorkload, IfcWorkstationRequirement, IfcWorkstationCompanyPolicy)*
- *IfcCost (used by IfcWorkstationCompanyPolicy)*
- *IfcDate (used in Att_ElectronicDocument, Att_DocumentType)*
- *IfcFixture (used by Att_TaskLighting)*
- *IfcInteger (used by IfcWorkstationWorkload, Att_Storage)*
- *IfcInteger (IfcWorkstationWorkload, Att_Storage, Att_PaperDocument, Att_DocumentType, Att_Specification)*
- *IfcLength (used by Att_SystemFurnitureType, Att_Worksurface, Att_Panel)*
- *IfcManufacturedElement (supertype of IfcSystemFurniture, used in IfcWorkstation)*
- *IfcOfficeEquipment (used by IfcWorkstation)*
- *IfcPolyCurve2D (used by IfcWorkstation, Att_Panel)*
- *IfcProductObject (superclass of IfcDocument, used by Att_PaperDocument)*
- *IfcReal (used in IfcWorkstationWorkload , Att_ElectronicDocument, Att_Drawing)*
- *IfcSpace (referenced in IfcSystemFurniture)*
- *IfcString (used in IfcWorkstationRequirement, IfcWorkstationCompanyPolicy, IfcWorkstation, Att_SystemFurnitureType, Att_Panel, Att_Worksurface, Att_Storage, Att_ElectronicDocument, Att_DocumentType, Att_Drawing, Att_Specification)*
- *IfcTime (used in Att_ElectronicDocument)*
- *IfcTimeDuration (used by IfcWorkstationWorkload)*
- *IfcTypeDefinition (used in IfcSystemFurniture, Panel, Worksurface, Storage, IfcDocument)*
- *IfcUnit (used in Att_Drawing)*

New object types required

- *IfcWorkstationWorkload ß subclass of IfcControlObject*
  - Average_workhour_weekly: IfcTimeDuration
  - Total_paperfiles_stored: IfcInteger
  - <used to determine file storage>
  - Average_paperfiles_produced_daily: IfcInteger
  - Total_computerfiles: IfcReal
  - <used to determine computer equipment>
  - <in unit of MB>
- *IfcWorkstationCompanyPolicy ß subclass of IfcControlObject*

- Employee_Type: IfcString
- <e.g. manager, programmer, secretary, etc.>
- Max_workstation_size: IfcArea
- Min_workstation_size: IfcArea
- Furniture_style: IfcString
- Cost_limit: IfcCost

• *IfcWorkstation ß subclass of IfcAssembledElement*
  - Components: Set [0:N] of Ref. to IfcManufacturedElement
  - <list of worksurfaces and storages, tables, chairs, etc., excluding the vertical panels>
  - Equipment: Set [0:N] of Ref. to IfcOfficeEquipment
  - <list of office equipment>
  - Panels: Set [0:N] of Ref. to IfcSystemFurniture
  - <list of furniture types, i.e. panel types>
  - Profile: IfcPolyCurve2D
  - Group: Ref. to IfcWorkstationGroup (to be defined in the next process)
  - <represents the workstation group that the workstation belongs to >
  - Has_boundingbox: IfcBoundingBox
  - Workload: IfcWorkstationWorkload
  - Company_policy: Ref. to IfcWorkstationCompanyPolicy
  - Assigned_to: Set [0:N] of Ref. to IfcActor
  - Zones: Set [0:N] of IfcWorkstationZone2D

• *IfcSystemFurniture ß subtype of IfcManufacturedObject*
  - Furniture_type: Ref. to IfcTypeDefinition
  - <Panel, Worksurface, Storage>
  - Workstation: Set [0:N] of Ref. to IfcWorkstation
  - Stored_in: Ref. to IfcSpace

• *Att_SystemFurnitureType ß shared AttributeSet of Furniture_type in IfcSystemFurniture*
  - Group_code: IfcString <e.g. panels, worksurfaces, storages, etc.>
  - Width: IfcLength <i.e. nominal width>
  - Height: IfcLength <i.e. nominal length>
  - Finishing: IfcString <e.g. walnut, fabric>

• *Panel ß of IfcTypeDefinition of Type in IfcSystemFurniture*
  target object: IfcSystemFurniture
  shared = Att_SystemFurnitureType
  occurance = Att_Panel

• *Worksurface ß of IfcTypeDefinition of FurnitureType in IfcSystemFurniture*
  target object: IfcSystemFurniture
  shared = Att_SystemFurnitureType
  occurance = Att_Worksurface

• *Storage ß of IfcTypeDefinition of FurnitureType in IfcSystemFurniture*
  target object: IfcSystemFurniture
  shared = Att_SystemFurnitureType
  occurance = Att_Storage

• *Att_Panel ß occurance AttributeSet for FurnitureType Panel in IfcSystemFurniture*
  - Shape: IfcPolyCurve2D
  - Opening: IfcPolyCurve2D
  - Panel_type: IfcString
  - <e.g. Acoustical, Horz_Seg, Monolithic, Glazed, Open, Ends, Door, Screen, etc.>
- Thickness: IfcLength

• *Att_Worksurface ß occurance AttributeSet for FurnitureType Worksurface in IfcSystemFurniture*

- Use_Purpose: IfcString
- <e.g. writing/reading, computer, meeting, printer, reference files, etc.>
- Support_type: IfcString
- <i.e. Freestanding or supported>
- Hunging_Height: IfcLength
- Thickness: IfcLength
- Shape_description: IfcString
- <corner square, rectangle, etc.>

• *Att_Storage ß occurance AttributeSet for FurnitureType Storage in IfcSystemFurniture*
  - IsOverhead: IfcBoolean (not sure whether IfcBoolean exists in the current version)
  - Support_type: IfcString
  - <i.e. Freestanding or supported>
  - Use_Purpose: IfcString
  - <e.g. shelf, stationary, office supplies, personal items, etc.>
  - Number_of_drawers: IfcInteger
  - Hunging_height: IfcLength <if IsOverhead>
  - Depth: IfcLength

• *Att_TaskLighting ß occurance AttributeSet in IfcFixture*
  - <<to be defined>>

• *IfcWorkstationRequirement ß subclass of IfcControlObject*
  - Security_Requirements: Set [0:N] of IfcString
  - Privacy_Requirements: Set [0:N] of IfcString
  - Special_Requirements: Set [0:N] of IfcString

• *IfcWorkstationZone2D ß used by IfcWorkstation*
  - <this class has no superclass>
  - Workstation_zonetype: IfcString
  - <e.g. worktask, cirrculation, chair_clearance, etc.>
  - Length: IfcLength
  - Width: IfcLength
  - In_workstation: Ref. to IfcWorkstation

• *IfcDocument ß subclass of IfcControlObject*
  ß subclass of IfcProductObject
  - GenericDocumentType: IfcTypeDefinition
  - <used to differentiate between an electronic and a paper document>
  - DocumentType: IfcTypeDefinition
  - <used to differentiate between a drawing and specification, and etc.>

• *ElectronicDocument ß of IfcTypeDefinition of GenericDocumentType in IfcDocument*
  target object = “IfcDocument”
  shared = <none>
  occurance = Att_ElectronicDocument

• *PaperDocument ß of IfcTypeDefinition of GenericDocumentType in IfcDocument*
  target object= “IfcDocument”
  shared = <none>
  occurance = Att_PaperDocument

• *Att_ElectronicDocument ß occurance AttributeSet for GenericDocumentType ElectronicDocument in IfcDocument*
  - File_name: IfcString
  - FileExtension_name: IfcString
  - Software: IfcString
  - File_size: IfcReal (in unit of KB)

- Directory: IftString
- Backup_file: Ref. to IfcDocument
- Paper_copy: Ref. to IfcDocument
- Last_save_time: IfcTime
- Last_save_date: IfcDate
- Type: IfcString <hiden, readonly, etc.)

- *Att_PaperDocument ß occurance AttributeSet for GenericDocumentType Paper Document in IfcDocument*
  - Location: Ref. to IfcProductObject <more appropriate if there is something like IfcRoot>
  - Total_pages: IfcInteger
  - Electronic_copy: Ref. to IfcDocument

- *Drawing ß of IfcTypeDefinition of DocumentType in IfcDocument*

  target object = “IfcDocument”

  shared = Att_DocumentType

  occurance = Att_Drawing

- *Specification ß of IfcTypeDefinition of DocumentType in IfcDocument*

  target object = “IfcDocument”

  shared = Att_DocumentType

  occurance = Att_Specification

- *Att_DocumentType ß shared AttributeSet for DocumentType in IfcDocument*
  - Author: IfcString
  - Company: IfcString
  - Title: IfcString
  - Revision_Code: IfcString
  - Revision_Number: IfcInteger
  - Last_modified_date: IfcDate
  - First_created_date: IfcDate

- *Att_Drawing ß occurance AttributeSet for DocumentType Drawing in IfcDocument*
  - Drawin_id: IfcString
  - Specifications: Set [0:N] of Ref. to IfcDocument
  - Scale: IfcReal
  - Unit: IfcUnit
  - Related_drawings: Set [0:N] of Ref. to IfcDocument

- *Att_Specification ß occurance AttributeSet for DocumentType Specificatioin in IfcDocument*
  - Specification_id: IfcString
  - General_description: IfcString
  - Related_drawings: Set [0:N] of Ref. to IfcDocument
  - Total_words: IfcInteger

- *IfcWorkstationGroup as defined in the next process “Floor Layout of Workstations”*

4.16.2.5 RoadMap Issues

Interoperability Issues

Applications from which information is needed:

- *Architecture*
- *CM*
- *HVAC*
- *Building Service*

Applications for which information is produced:

- *Architecture*
- *CM*
- *Building Service*

Value of software supporting this process

- *Facilities Managers/Building Owners: Very High (in the top 3)*
- *Architecture: High (in the top 5)*
- *CM/Cost Est.: Very High (in the top 3)*
    - *have indicated that the FM space and systems furniture information would be very useful for building remodeling.*
- *Building Service: Very High (in the top 3)*
- *HVAC:*

Sponsor Software Companies

Software Companies that have shown an interest in developing applications which implement the process

- *Naoki Systems Inc.*

4.16.2.6 Issues idendified in reviews

{{ Reviewing group - Reviewed for: }}

- ***Issues:***
    - ***{{ Proposed resolution }}***
- **{{ Issue 1 }}**
    - *{{ Proposed resolution }}*
- **{{ Issue 2 }}**
    - *{{ Proposed resolution }}*

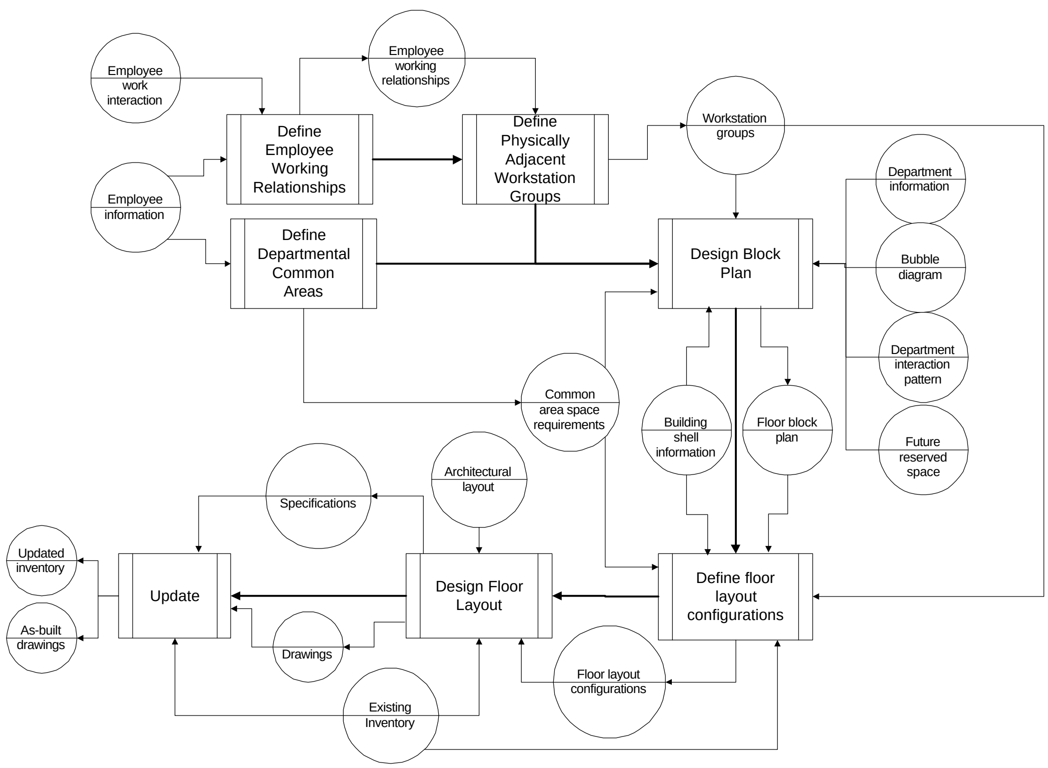
4.16.3 Floor Layout of Workstations for an Open Office

The facility manager (also interior designers, architects, or furniture dealers, etc.) designs the layout of the workstations for an open office. This process is part of the entire floor furniture and equipment planning for the department(s), and occurs after typical individual workstations have been designed. CAD-based computer programs can automate the layout design process and result in cohesive, productive and suitable department offices.

4.16.3.1 Industry Process Definition

<< refer to usage scenario >>

4.16.3.2 Process Diagram



4.16.3.3 Process Analysis

4.16.3.3.1 Define Employee Working Relationships

Define the individual employees working interaction patterns and meeting frequencies according to the work they perform.

Input Information:

- *Employee Information*
  - *employee name and title*
  - *work types*
  - *employee roles - roles contain requirements for interactivity*
- *Employee work interaction*
  - *within department and outside department*
    - *with whom (or location)*

- *daily frequency*
- *average duration each meeting*

Output Information:

- *Employee working relationships (in the form of table)*
  - *department title*
  - *interaction pattern summary*

Project Model Usage Requirements:

Existing Classes:

- *IfcControlObject (superclass of IfcInteraction)*
- *IfcTypeDefinition (used in IfcInteraction)*
- *IfcInteger (used in IfcInteraction)*
- *IfcTimeDuration (used in IfcInteraction)*
- *IfcString (used in IfcInteraction, Att_ActorInteraction)*
- *IfcActor (referenced in Att_ActorInteraction)*

New Classes:

- *IfcInteraction ‹ subclass of IfcControlObject*
  - Description: IfcString
  - Interaction_type: IfcTypeDefinition
  - Frequency_daily: IfcInteger
  - Average_duration: IfcTimeDuration
- *ActorInteraction ‹ of IfcTypeDefinition of Interaction_type in IfcInteraction*
  target object = "IfcInteraction"
  shared = <none>
  occurance = Att_ActorInteraction
- *Att_ActorInteraction ‹ occurance AttributeSet for Interaction_type ActorInteraction in IfcInteraction*
  - Actors: Set [2:N] of Ref. to IfcActor
  - Locations: IfcString

4.16.3.3.2 Define Physically Adjacent Workstation Groups

Define the functional workstation groups according to the individual employees working relationships. A group consists of one or a few different types of typical workstations that have close working relationships, frequent interactions, and perform the same kind of function.

Input Information:

- *Employee working relationships*

Output Information:

- *Workstation groups*
  - *Workstation group function name (e.g. programmer 1, marketing 2, etc.)*
  - *List of workstations*
  - *workstation group relationships (e.g. probably bubble diagram, or interaction pattern table)*

Project Model Usage Requirements:

Existing Classes:

- *IfcAssembledElement (superclass of IfcWorkstationGroup)*
- *IfcString (used in IfcWorkstationGroup)*

New Classes:

- *IfcWorkstationGroup ‹ subclass of IfcAssembledElement*
  - Functional_name: IfcString
  - Workstations: Set [0:N] of Ref. to IfcWorkstation

4.16.3.3.3 Define Departmental Common Areas

Define the areas that are shared by all employees in the department, such as common circulation and conference rooms, etc.

Input Information:

- *minimum standards from company policy or architectural group:*
- *Employee Information*
  - *types, total numbers*
  - *Work types*

Output Information:

- *Common area space requirements*
  - *circulation space*
  - *coffee room space*

Project Model Usage Requirements:

Existing Classes:

- *IfcActor (referenced by IfcWorkstationGroup.Department)*

New Classes:

- *IfcWorkstationGroup*
  - <all attributes described in the previous steps + following>
  - Department: Ref. to IfcActor

4.16.3.3.4 Design Block Plan

Segment large spaces for workstation groups according to the relationships between the workstation groups, and the relationships between departments in case of multiple departments. Floor geometry constraints such as column grids, ceiling grids, window grids, the space footage must be taken into consideration. A floor block can contain one or more workstation groups, or one or more workstations.

Input Information:

- *Departmental information*
  - *name, function*
- *Bubble diagram*
  - *e.g. for interdepartmental interactions*
- *Department interaction patterns*
  - *all department names, sizes*
  - *interaction leveling (primary, secondary, tertiary, insignificant, and none)*
  - *interaction directions*
- *Common area space requirements*
- *Workstation groups*
- *Building shell information*
  - *walls*

- *column grids*
- *ceiling grids*
- *window grids*
- *floor openings (e.g. access for raised floor)*
- *space openings (e.g. exit area)*
- *building core (e.g. elevators, restroom, etc.)*
- *space footage*
- *Future reserved space*
  - *footage*
  - *preferred locations*
  - *least area requirements*

Output Information:

- *Floor block plan*
  - *floor blocks*
    - *block id, block name, block footage*
    - *owning department*
    - *list of contained workstation groups*
    - *list of contained workstations*

Project Model Usage Requirements:

Existing Classes:

- *IfcArea (used in IfcFloorBlock)*
- *IfcAssembledElement (superclass of IfcFloorBlock)*
- *IfcPolyCurve2D (used in IfcFloorBlock)*
- *IfcString (used by IfcFloorBlock, Att_SpaceType)*
- *IfcSpace (referenced in IfcFloorBlock)*
- *SpaceType (extended by Att_SpaceType)*

New Classes:

- *IfcFloorBlock ‹ subclass of IfcAssembledElement*

  {{need to varify this with Architecture Domain}}
  - Function_name: IfcString
  - Workstation_groups: Set [0:N] of Ref. to IfcWorkstationGroup
  - In_space: Ref. to IfcSpace
  - Profile: IfcPolyCurve2D
  - Area: IfcArea
- *Att_SpaceType ß shared AttributeSet for TypeDefinition SpaceType defined in R1.0*
  - Space_name: IfcString
  - General_description: IfcString
  - Space_catalog: IfcString

4.16.3.3.5 Define Floor Layout Configurations

Define all the detailed footage of all the workstations, workstation groups and departmental boundaries.

Input Information:

- *Floor block plan*

- *Common area space requirements*
- *Workstation groups*
- *Building shell information*
- *Existing inventory*
    - *furniture in store rooms*

Output Information:

- *Floor layout configurations*
    - *workstations*
        - *id, name, owning department, and user*
        - *footage*
    - *workstation groups*
    - *departmental boundaries*

Project Model Usage Requirements:

Existing Classes:

- *IfcSpace (referenced in IfcWorkstationGroup)*
- *IfcArea (referenced in IfcWorkstationGroup)*
- *IfcPolyCurve2D (referenced in IfcWorkstationGroup)*
- *IfcManufacturedElement (referenced by IfcFloorBlock, etc.)*
- *IfcOfficeEquipment (referenced by IfcWorkstationGroup)*

New Classes:

- *IfcWorkstationGroup ß extended from first step*
    - {{all items described in previous steps + the following}}
    - In_floor_block: Ref. to IfcFloorBlock
    - In_space: Ref. to IfcSpace
    - Profile: IfcPolyCurve2D
    - Total_area: IfcArea
    - Shared_furniture: Set [0:N] of Ref. to IfcManufacturedElement
    - <shared furniture is not part of any workstations in the workstation group, e.g. a table for supporting a shared printer>
    - Shared_equipment: Set [0:N] Ref. to IfcOfficeEquipment
    - <shared equipment is not part of any workstations in the workstation group, e.g. a shared printer>
    - Workstation_groups: Set [0:N] of Ref. to IfcWorkstationGroup
    - <a workstation group can contain other groups>

4.16.3.3.6 Design Floor Layout

Produce the workstation layout drawings and define the specifications.

Input Information:

- *Floor layout configurations*
- *Existing inventory*
- *Architectural layout*

Output Information:

- *Floor layout drawings*
    - *drawing id, name, author, proof, company, etc.*
- *Floor layout specifications*

- *Materials*
- *Installation requirements*

Project Model Usage Requirements:

Existing Classes:
<none in this step>

New Classes:

- *IfcDocument as defined in the last step of the process "Design of Workstations"*

4.16.3.3.7 Update

This is an on-going process that occurs during the course of design implementation. Inventories are updated. Drawings are changed and as-built drawings are produced overtime.

Input Information:
- *Existing inventory*
- *Floor layout drawings*
- *Floor layout specifications*

Output Information:
- *As-built drawings*
- *Updated inventories*

Project Model Usage Requirements:

Existing Classes:

- *<none in this step>*

New Classes:

- *IfcSpaceInventory (as defined in the 1st process: Occupancy Planning)*
- *IfcFurnitureInventory (as defined in the 1st process: Occupancy Planning)*
- *IfcEquipmentInventory (as defined in the 1st process: Occupancy Planning)*

4.16.3.4 IFC Model Impact

Usage/Extensions to R1.0 object types

- *IfcActor (referenced in Att_ActorInteraction, IfcWorkstationGroup)*
- *IfcArea (used in IfcWorkstationGroup, IfcFloorBlock)*
- *IfcAssembledElement (superclass of IfcWorkstationGroup, IfcFloorBlock)*
- *IfcControlObject (superclass of IfcInteraction)*
- *IfcOfficeEquipment (referenced in IfcWorkstationGroup)*
- *IfcManufacturedElement (referenced in IfcWorkstationGroup)*
- *IfcInteger (used in IfcInteraction)*
- *IfcPolyCurve2D (used in IfcWorkstationGroup, IfcFloorBlock)*
- *IfcSpace (referenced in IfcWorkstationGroup, IfcFloorBlock)*

- *IfcString (used in IfcInteraction, Att_ActorInteraction, IfcWorkstationGroup, IfcFloorBlock, Att_SpaceType)*
- *IfcTimeDuration (used in IfcInteraction)*
- *IfcTypeDefinition (used in IfcInteraction)*

New object types required

- *IfcInteraction ‹ subclass of IfcControlObject*
  - Description: IfcString
  - Interaction_type: IfcTypeDefinition
  - Frequency_daily: IfcInteger
  - Average_duration: IfcTimeDuration
- *ActorInteraction ‹ of IfcTypeDefinition of Interaction_type in IfcInteraction*
  target object = “IfcInteraction”
  shared = <none>
  occurance = Att_ActorInteraction
- *Att_ActorInteraction ‹ occurance AttributeSet for Interaction_type ActorInteraction in IfcInteraction*
  - Actors: Set [2:N] of Ref. to IfcActor
  - Locations: IfcString
- *IfcWorkstationGroup ‹ subclass of IfcAssembledElement*
  - Functional_name: IfcString
  - Workstations: Set [0:N] of Ref. to IfcWorkstation
  - Department: Ref. to IfcActor
  - In_floor_block: Ref. to IfcFloorBlock
  - In_space: Ref. to IfcSpace
  - Profile: IfcPolyCurve2D
  - Total_area: IfcArea
  - Shared_furniture: Set [0:N] of Ref. to IfcManufacturedElement
  - <shared furniture is not part of any workstations in the workstation group, e.g. a table for supporting a shared printer>
  - Shared_equipment: Set [0:N] Ref. to IfcOfficeEquipment
  - <shared equipment is not part of any workstations in the workstation group, e.g. a shared printer>
  - Workstation_groups: Set [0:N] of Ref. to IfcWorkstationGroup
  - <a workstation group can contain other groups>
- *IfcFloorBlock ‹ subclass of IfcAssembledElement*
  - Function_name: IfcString
  - Workstation_groups: Set [0:N] of Ref. to IfcWorkstationGroup
  - In_space: Ref. to IfcSpace
  - Profile: IfcPolyCurve2D
  - Area: IfcArea
- *Att_SpaceType ß shared AttributeSet for TypeDefinition SpaceType defined in R1.0*
  - Space_name: IfcString
  - General_description: IfcString
  - Space_catalog: IfcString
- *IfcDocument as defined in the last step of the process “Design of Workstations”*
- *IfcWorkstation as defined in the process “Design of Workstations”*
- *IfcSpaceInventory (as defined in the 1st process: Occupancy Planning)*

- *IfcFurnitureInventory (as defined in the 1st process: Occupancy Planning)*
- *IfcEquipmentInventory (as defined in the 1st process: Occupancy Planning)*

4.16.3.5 RoadMap Issues

Interoperability Issues

Applications from which information is needed:

- *Architecture*
- *CM*
- *HVAC*
- *Building Service*

Applications for which information is produced:

- *Architecture*
- *CM*
- *Building Service*

Value of software supporting this process

- *Facilities Managers/Building Owners: Very High (in the top 3)*
- *Architecture: High (in the top 5)*
- *CM/Cost Est.: Very High (in the top 3)*
  - *have indicated that the FM space and systems furniture information would be very useful for building remodeling.*
- *Building Service: Very High (in the top 3)*
- *HVAC:*

Sponsor Software Companies

Software Companies that have shown an interest in developing applications which implement the process

- *Naoki Systems Inc.*

4.16.3.6 Issues idendified in reviews

{{ Reviewing group - Reviewed for: }}

- ***Issues:***
  - ***{{ Proposed resolution }}***
- **{{ Issue 1 }}**
  - *{{ Proposed resolution }}*
- **{{ Issue 2 }}**
  - *{{ Proposed resolution }}*

Simulation

4.17 SI-1 Visualization

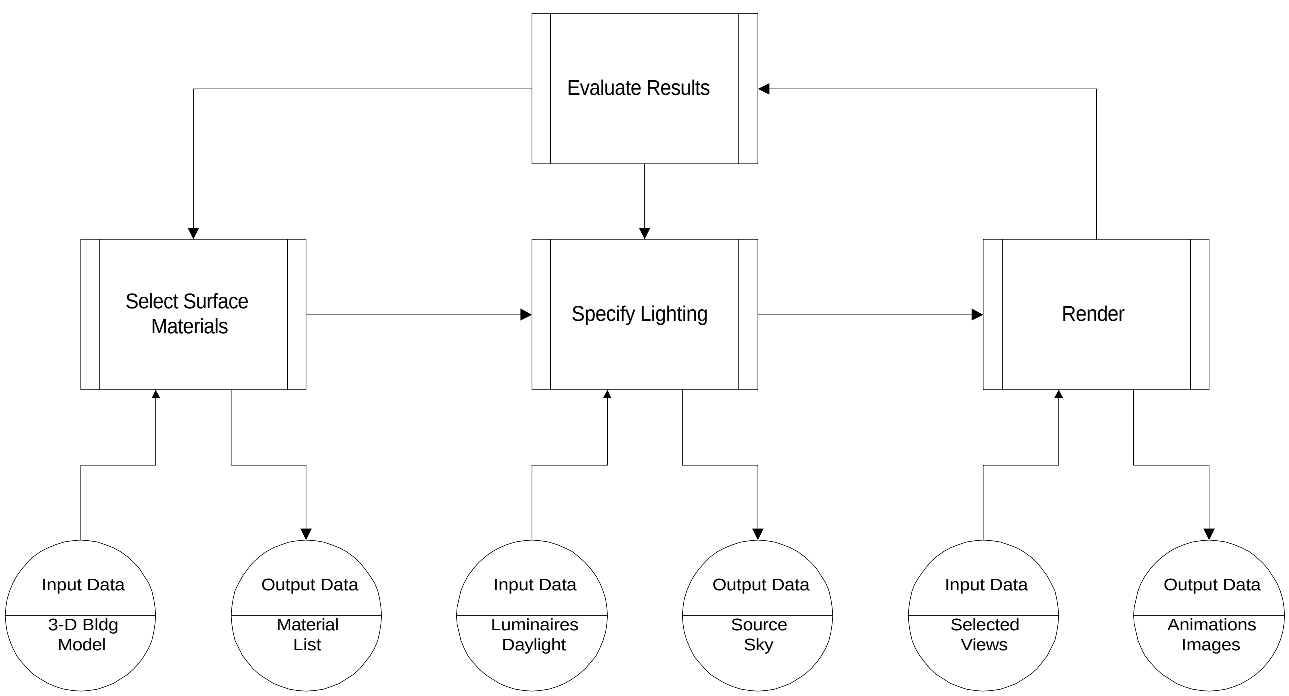
4.17.1 Architectural Visualization

Visualization is performed by architects, lighting engineers and renderers with computer and electronic visualization skills. It can be used at any point in the building, lighting or interior design process, as well as during the occupancy of the building. Three-dimensional representation of a space or a building is the starting point; information about surface materials and sources of light is required for the simulation. The former can be obtained from manuals, manufacturers' catalogues, databases, etc. The later is available in technical literture and from specialised computer models. The resulting images (renderings or animations) and data (luminance) can be used for many purposes: communication about the "looks" of a design solution to making design and engineering decisions.

4.17.1.1 Industry Process Definition

In the design of a building or other structure, the architect or designer may want to see what the building or the structure will look like, or may want to render images for the client's benefit. Such visualization may be desired at any time from the earliest architectural design or retrofitting to the final interior design. Visualization is the key to solving lighting and daylighting design problems, and is also important in assessing building performance and human comfort issues. IFC support of this process may reduce input preparation time by 75-85% process (through automatic acquisition of building geometry and all surface properties) and thus make the use of the corresponding applications economically feasible.

4.17.1.2 Process Diagram



4.17.1.3 Process Analysis

Select Surface Materials

Select materials to associate with building surfaces, and define material properties such as reflectance, transmittance, colors, patterns, texture, etc. Input is the three-dimensional building model; output are the materials associated with each modeled surface or solid.

Input Information:

- *Three-dimensional building model*

Output Information:

- *Materials associated with each 3-d surface (list)*

Specify Lighting

In order to perform a valid visual simulation, it is necessary to select and place light sources (luminaires) and specify daylight conditions. Luminaires may be selected from manufacturer's catalogs, and the sun and sky conditions may be taken from a set of quantitative models appropriate to the building site.

Input Information:

- *Light source configuration and distribution data*
- *Daylight models for this site*

Output Information:

- *Positions and types of light sources (drawings and specs)*
- *Sky distribution, solar position (fromtime of day and year)*

Rendering

Compute 2-dimensional images for visual evaluation.

Input Information:

- *Selected views for this model*
- *Animation path (optional)*

Output Information:

- *Two-dimensional color images (floating point)*
- *Luminance and isolux contour plots (optional)*
- *Animations (optional)*

Evaluate Results

Once one or more images have been produced, it is often desirable to go back and iterate on the material selection and/or light source selection and placement

Project Model Usage Requirements:

Existing Classes: all that define the geometry of the space or building in the simulation

New Classes:

- Light source

- Surface

4.17.1.4 IFC Model Impact

Usage/Extensions to R1.0 object types

- *IfcMaterialLayer*
  - *bidirectional scattering distribution function (BSDF) or model thereof includes: spectral reflectance and transmittance, specularity and roughness*
  - *polarization properties*

New object types required

- *IfcLightSource*
  - *Spectral powerdistribution (lamp)*
  - *Luminaire geometry*
  - *Photometric output distribution*
- *IfcSurface*
  - *General shape (e.g., polygon, sphere, etc.)*
  - *Dimensions*
  - *Material and parameterization*

4.17.1.5 RoadMap Issues

Interoperability Issues

Applications from which information is needed:

- *CAD software*

Applications for which information is produced:

- *to be determined*

Value of software supporting this process

{{In this section, please allow for the other domains to rank your process in order of precedance for their domain, this allows us to examine the issue on a group as well as an individual level}}

- *{{ discipline 1 }} - {{value from 1-10, 1 being the highest value, 10 being the lowest]]*
- *{{ discipline 2 }} - {{value from 1-10]]*

Sponsor Software Companies

Software Companies that have shown an interest in developing applications which implement the process

- *LBNL (Radiance 3.0)*
- *Lightscape Technologies*
- *Arris Integra*

- *3DStudio (rendering)*
- *Pixar (Renderman)*
- *Lightworks*
- *others*

4.17.1.6 Issues idendified in reviews

{{ Reviewing group - Reviewed for: }}

Issues:

- ***{{ Proposed resolution }}***

- **{{ Issue 1 }}**
  - *{{ Proposed resolution }}*
- **{{ Issue 2 }}**
  - *{{ Proposed resolution }}*

Structural Engineering

4.18 ST-1 Steel Frame Structures

{{ Model Requirements Analysis for this project not yet available }}

4.19 ST-2 Concrete Frame Structures

{{ This project has been delayed for inclusion in IFC Release 3.0 }}

4.20 ST-3 Sub Structure Design

{{ This project has been delayed for inclusion in IFC Release 3.0 }}

4.21 ST-4 Structural Loads Definition

{{ Model Requirements Analysis for this project not yet available }}

Cross Domain Projects

4.22 XM-1 Referencing External Libraries

{{ This project has been delayed for inclusion in IFC Release 3.0 }}

4.23 XM-2 Project Document Management

4.23.1 Project Document Management

Project Document Management refers to all information pertaining to the documents used to estimate, bid, purchase, and manage the building process as well as for use within the Facilities Management domain. This data identifies the document, the author of the document, changes to the document since the last change, and relationships to other documents.

- *Who performs this process? All software vendors that use drawings, specifications, and sketches during the life cycle of a project. This would include CAD, estimating, scheduling, management, and facilities management software vendors.*
- *When in the project lifecycle it is performed? From the very inception of the project, where these documents are used to define the project, through the construction of the project with all of its changes, through the management of the "building" once the project is complete.*
- *What other processes does it relate to (input from/output to/controlled by)? This process starts in the creation and modification of the documents and outputs to all processes that use the documents as a means of identification. This would include estimating where changes to the work are usually quantified by document, management, where the documents are used to control the flow of work on a project and establish what is being built by document, and Facilities Management, where documents are the prime method of identifying actual conditions in a facility.*

4.23.1.1 Industry Process Definition

For Contract Drawings and Sketches, the Architect start this process during the creation of the drawings by entering information regarding the drawings. This information would include:

- Document type
- Document Id
- Description
- Document Date
- Revision Number
- Revision Date
- Document Type and Id of related documents. This might include relationships between drawings and sketches, or even objects in the drawings with objects in a sketch or maybe the object as identified within a specification.
- Document Author
- Document Revision Author
- Bulletin/Addenda reference
- Related Documents, Sections, Details, Objects

In addition, the creation of objects onto the drawing will also trigger the saving of information regarding the objects. This process is handled by the CAD software. The information saved would include:

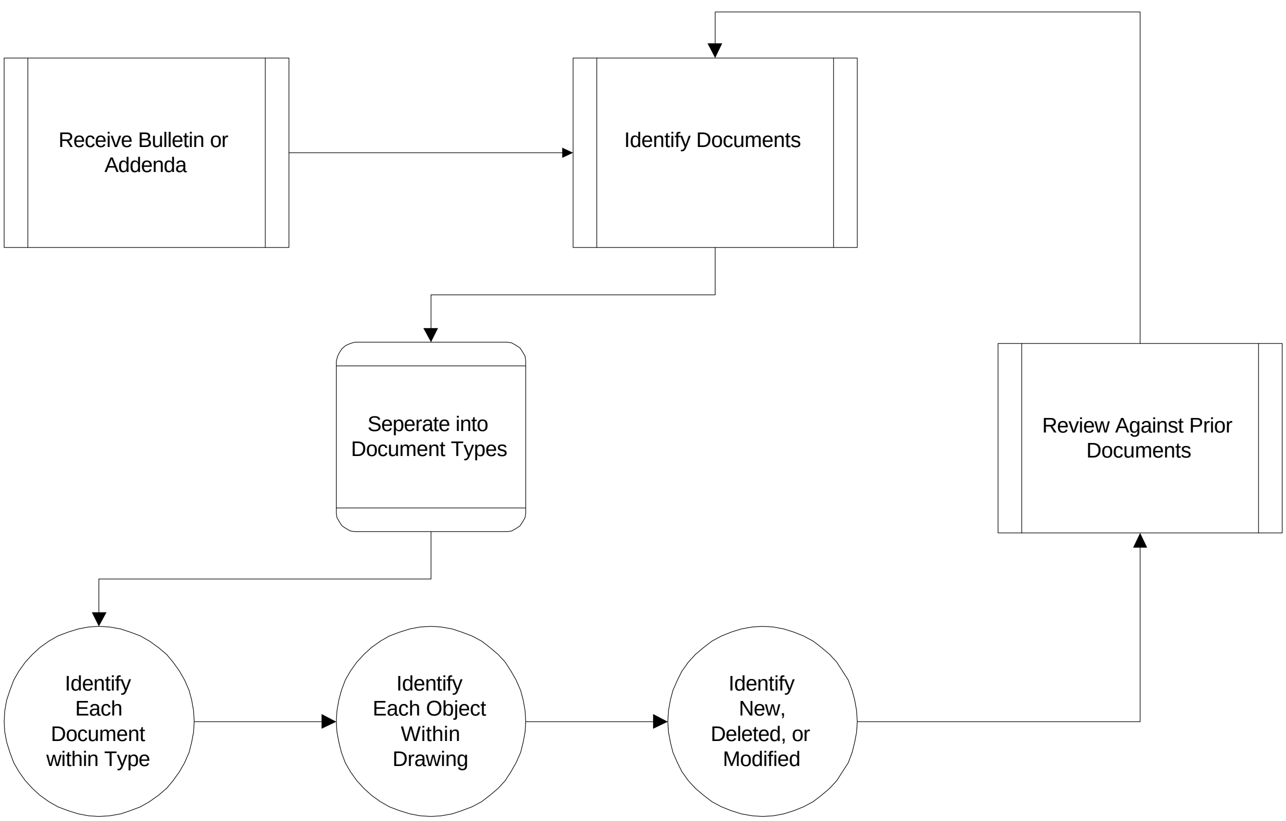
The List of Objects

- Original
- Added
- Deleted
- Modified
- Dates for all of these
- Revisions for all of these
- Possible relationships between the objects of this drawing and the objects on other drawings, specifications, and sketches created by the Architect.

Similar information would be required for implementation of the same for Specifications. These will not be modeled here.

Once the information is provided to the document, software using drawings can take advantage of the information to organize the processes of change throughout a project as well as the interconnection of information between project contract documents, such as between drawings, drawing sections, drawing details, sketches, and specification sections.

4.23.1.2 Process Diagram



4.23.1.3 Process Analysis

Identifications Supplied By Author

The Architect or author of the document provides information regarding the document he/she is creating.

Input Information:

- *The document type, whether it is a drawing, specification, sketch, etc.*
- *The document details such as Drawing Number, Drawing Date, Author, etc.*

Project Model Usage Requirements:

To be determined

Identifications Supplied by Vendor

The CAD or Specifications Software Vendor (or Author?)would assign an identification and information regarding those objects within the document.

Input Information:

- *Object Types*
- *Object Specific information such as Object Identification, Creation Date, Author, etc.*

Project Model Usage Requirements:

To be determined

Document Modifications

The CAD or Specifications Software Vendor (or Author?) would keep track of modifications made to the documents with respect to Revision and date.

Input Information:

- *Type of Modification, such as created, modified, deleted*
- *Modification Details, such as Creation Date, Author of Change, Change Identifications*

Project Model Usage Requirements:

To be determined

Identify Relationships with Other Documents

A link is then made to the appropriate documents, where information is contained not within the current document.

Input Information:

- *Document Type*
- *Document Identification*
- *Internal Document Reference*

output Information:

- *Document Identification*
- *Document Type*
- *Internal Document Reference*
- *Change Information*

Project Model Usage Requirements:

To be determined

Estimating

Estimating software packages can now use the information provided above to estimate changes to the project on a document by document and change by change basis. This would include changes made over multiple documents since the change identification can be identical between documents.

output Information:

- *Document Type*
- *Document Identification*
- *Internal Document Reference*
- *Change Information*

Project Model Usage Requirements:

To be determined

Scheduling

Scheduling software packages can now use the information provided above to estimate changes to the project schedule on a document by document and change by change basis. This would include changes made over multiple documents since the change identification can be identical between documents.

output Information:

- *Document Type*
- *Document Identification*
- *Internal Document Reference*
- *Change Information*

Project Model Usage Requirements:

To be determined

Project Management

Project Management software packages can now use the information provided above to estimate changes to the project on a document by document and change by change basis. This would include changes made over multiple documents since the change identification can be identical between documents.

output Information:

- *Document Type*
- *Document Identification*
- *Internal Document Reference*
- *Change Information*

Project Model Usage Requirements:

To be determined

4.23.1.4 IFC Model Impact

Extensions to R1.0 object types

To be determined

New object types

Document Type Object
Document Object

4.23.1.5 RoadMap Issues

Interoperability issues

Disciplines from which information is needed:

- *Architects*
- *CAD Software*
- *Engineers (those who create contract documents)*
- *Facilities Management*
- *Specifications Software Vendors*

Disciplines for which information is produced:

- *Owners*
- *Architects*
- *Engineers*
- *Construction Professionals*
- *Estimators*
- *Purchasers*
- *Facilities Management*

Value of software supporting this process

- *Construction Professionals {I consider this my highest priority - a definite 10}*

Software Vendors willing to participate

- *Frontrunner, LLC*
- *Turner Corporation Internal Development*
- *Autodesk*
- *Bentley Systems (preliminary interest)*

4.24 XM-3 IFC Model - Enabling Mechanisms

{{ Model Requirements Analysis for this project not yet available }}

*** Instructions to R2.0 Project Teams: ***

Writing "Model Requirements Analysis" documents

This document is provided as a formatting template for the documentation of you analysis of the information requirements for each of the tasks you defined in your "AEC Industry Processes and Usage Scenaria". Note that this document should reference these process tasks as already defined and not include a redundant definition of process tasks.

The document is structured so that several process analyses can be combined into a single integrated document for all R2.0 projects.

Therefore ***please do not modify the document structure***.

Document structure is roughly:

Heading 1	1.	Project Document TYPE header
Heading 2		Industry Domain
Heading 3	1.1	Project header
Heading 4	1.1.1	Process header
Heading 5	1.1.1.3	Process Analysis
Heading 6	1.1.1.3.1	Process task definition and analysis

Each process analysis is presented within a “Heading 4” block (listed as “1.1.1 {{ Process/Functionality Name }} “). Below this, within each "Heading 5" block (listed as "1.1.1.1 {{Process task A}}"), a process task is defined and then analyzed in terms of its required input and resulting output information.

Notes:

1. The process definitions in this document must be consistent with their presentation in the companion project document → "AEC Industry Processes and Usage Scenaria"
2. Note that anything enclosed in double curly braces → {{ xxx }} ← are instructions to you (the writer). You must replace these with content as described in the instructions.
3. We are using the TQM diagramming template in Visio 4.0 for the process diagrams. A sample Visio 4.0 TQM diagram is included → just double-click to edit

   Note that you should include all of the primary tasks in your process and identify the informational requirements for each task. Details for these informational inputs/outputs are imperative to completion of the requirements definition to be used by the Information Modeler assigned to design the model extensions to support your project.
4. Please see the example file sent with this template → R2an_EX1.doc.
5. Naming convention for your "Model Requirements Analysis" -- R2an_XXn.doc (where XXn is the Proj ID in the first column of the spreadsheet). Example: "Completion of the Architectural Model" has an ID of AR-1 --- thus the filename will be "R2an_AR1.doc".
6. This is a project 'working' document. It will contribute content to the "Model Requirements Analysis" section of the Release 2.0 Specifications volume II -- IFC Object Model for AEC Projects.

7. The process definition sections from this document must be consistent with those in the "AEC Industry Processes and Usage Scenaria" .
8. This template extends previous versions of the template for this document in multiple ways, including:
    - *AEC Domain groupings have been added as "Heading 2" which demotes previous headings 2, 3, and 4*
    - *Process Tasks are not "Heading 6" sections. This will enable cross document references to specific process tasks*
    - *Model Requirements Analysis sections for each process task expanded to include behavior.*

If you have questions please send email!

Regards,

Richard See

4.25 IFC Model Requirements Analysis Template {{ProjID}} - {{ Project Name }}

4.25.1 {{ Process/Functionality name }}

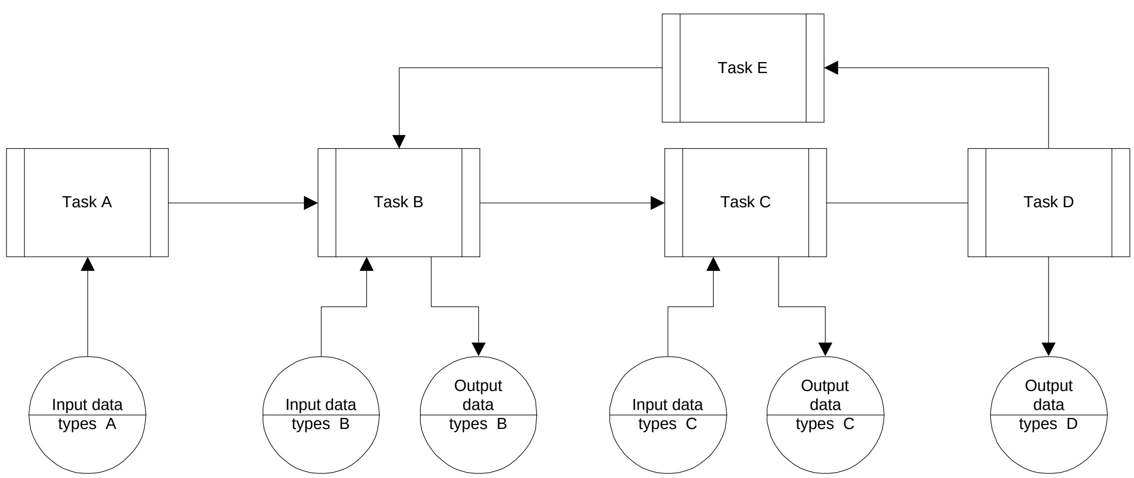
{{ provide any explanation you think would help the reader understand }}

- *who performs this process*
- *When in the project lifecycle it is performed*
- *What other processes it relates to (input from/output to/controlled by) }}*

4.25.1.1 Industry Process Definition

{{ provide a clear, concise definition of this process/functionality in terms that other AEC industry professionals will understand }}

4.25.1.2 Process Diagram



4.25.1.3 Process Analysis

{{ complete an outline of the tasks you envision in software supporting this industry process }}

4.25.1.3.1 {{ Process task A }}

{{ description of this process task }}

Input Information:

- *{{ object type 1 }}*
- *{{ object type 2 }}*

output Information:

- *{{ object type 1 }}*
- *{{ object type 2 }}*

Project Model Usage Requirements:

Existing Classes:

- ***{{ Object type name }}***

  Data

  - *{{ Data description à type }}*
    - *{{ notes }}*

  Behavior

  - *{{ Behavior description }}*
    - *{{ notes }}*

New Classes:

- ***{{ Object type name }}***

  Data

  - *{{ Data description à type }}*
    - *{{ notes }}*

  Behavior

  - *{{ Behavior description }}*
    - *{{ notes }}*

4.25.1.3.2 {{ Process task B }}

{{ description of this process task }}

Input Information:

- *{{ object type 1 }}*
- *{{ object type 2 }}*

output Information:

- *{{ object type 1 }}*
- *{{ object type 2 }}*

Project Model Usage Requirements:

Existing Classes:

- ***{{ Object type name }}***

  Data

- *{{ Data description à type }}*
    - *{{ notes }}*

Behavior

- *{{ Behavior description }}*
    - *{{ notes }}*

New Classes:

- ***{{ Object type name }}***

  Data
  - *{{ Data description à type }}*
    - *{{ notes }}*

  Behavior
  - *{{ Behavior description }}*
    - *{{ notes }}*

4.25.1.3.3 {{ Process task C }}

{{ description of this process task }}

Input Information:

- *{{ object type 1 }}*
- *{{ object type 2 }}*

output Information:

- *{{ object type 1 }}*
- *{{ object type 2 }}*

Project Model Usage Requirements:

Existing Classes:

- ***{{ Object type name }}***

  Data
  - *{{ Data description à type }}*
    - *{{ notes }}*

  Behavior
  - *{{ Behavior description }}*
    - *{{ notes }}*

New Classes:

- ***{{ Object type name }}***

  Data
  - *{{ Data description à type }}*
    - *{{ notes }}*

  Behavior
  - *{{ Behavior description }}*
    - *{{ notes }}*

4.25.1.3.4 {{ Process task D }}

{{ description of this process task }}

Input Information:

- *{{ object type 1 }}*
- *{{ object type 2 }}*

output Information:

- *{{ object type 1 }}*
- *{{ object type 2 }}*

Project Model Usage Requirements:

Existing Classes:

- ***{{ Object type name }}***
  Data
  - *{{ Data description à type }}*
    - *{{ notes }}*

  Behavior
  - *{{ Behavior description }}*
    - *{{ notes }}*

New Classes:

- ***{{ Object type name }}***
  Data
  - *{{ Data description à type }}*
    - *{{ notes }}*

  Behavior
  - *{{ Behavior description }}*
    - *{{ notes }}*

4.25.1.3.5 {{ Process task E }}

{{ description of this process task }}

Input Information:

- *{{ object type 1 }}*
- *{{ object type 2 }}*

output Information:

- *{{ object type 1 }}*
- *{{ object type 2 }}*

Project Model Usage Requirements:

Existing Classes:

- ***{{ Object type name }}***
  Data
  - *{{ Data description à type }}*
    - *{{ notes }}*

  Behavior
  - *{{ Behavior description }}*
    - *{{ notes }}*

New Classes:

- ***{{ Object type name }}***

Data
- *{{ Data description à type }}*
    - *{{ notes }}*

Behavior
- *{{ Behavior description }}*
    - *{{ notes }}*

4.25.1.4 IFC Model Impact

This section summarizes the model requirements from all the process tasks analyzed above into two groups à Extensions to R1.0 model object types and proposed new object types for R2.0.

Usage/Extensions to R1.0 object types

- ***{{ Object type name }}***

  Data
  - *{{ Data description à type }}*
    - *{{ notes }}*

  Behavior
  - *{{ Behavior description }}*
    - *{{ notes }}*

- ***{{ Object type name }}***

  Data
  - *{{ Data description à type }}*
    - *{{ notes }}*

  Behavior
  - *{{ Behavior description }}*
    - *{{ notes }}*

New object types required

- ***{{ Object type name }}***

  Data
  - *{{ Data description à type }}*
    - *{{ notes }}*

  Behavior
  - *{{ Behavior description }}*
    - *{{ notes }}*

- ***{{ Object type name }}***

  Data
  - *{{ Data description à type }}*
    - *{{ notes }}*

  Behavior
  - *{{ Behavior description }}*
    - *{{ notes }}*

4.25.1.5 RoadMap Issues

Interoperability Issues

Applications from which information is needed:

- *{{ application 1 }}*
- *{{ application 2 }}*

Applications for which information is produced:

- *{{ application 1 }}*
- *{{ application 2 }}*

Value of software supporting this process

{{In this section, please allow for the other domains to rank your process in order of precedance for their domain, this allows us to examine the issue on a group as well as an individual level}}

- *{{ discipline 1 }} - {{value from 1-10, 1 being the highest value, 10 being the lowest]]*
- *{{ discipline 2 }} - {{value from 1-10]]*

Sponsor Software Companies

Software Companies that have shown an interest in developing applications which implement the process

- *{{ company 1 }}*
- *{{ company 2 }}*

4.25.1.6 Issues idendified in reviews

{{ Reviewing group - Reviewed for: }}

- ***Issues:***
  - ***{{ Proposed resolution }}***
- **{{ Issue 1 }}**
  - *{{ Proposed resolution }}*
- **{{ Issue 2 }}**
  - *{{ Proposed resolution }}*